

repository.ub.ac.id

**IMPLEMENTASI SISTEM MONITORING KEAMANAN RUMAH  
PINTAR BERBASIS *ALEXA VOICE COMMAND* PADA  
RASPBERRY Pi**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun Oleh :  
Bramantyo Ardi  
NIM : 145150300111102



PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Rumah Pintar atau biasa disebut *Smart home* merupakan sebuah rumah yang menyediakan kenyamanan dan kemudahan kepada penghuninya dalam pengontrolan peralatan di setiap waktunya (Tri Fajar dkk., 2009). Sehingga semua fungsi rumah tangga seperti *rice cooker*, TV, radio, kulkas, dan AC dapat dikendalikan dengan menggunakan remote kontrol (nirkabel). Bahkan di era maju seperti saat ini, perangkat dan perabot rumah tangga dapat dikendalikan melalui internet, *smartphone* dan aplikasi-aplikasi pemrograman (software) lainnya. Beberapa tahun belakangan ini, banyak riset dan pengembangan teknologi dalam bidang *smart home*. Salah satu pengembangannya adalah pengontrolan perangkat *smart home* menggunakan suara. Pengontrolan menggunakan suara memberikan kemudahan dan kenyamanan bagi pengguna. Salah satu teknologi yang sedang digunakan saat ini adalah *voice control system* yang dikembangkan oleh Amazon yaitu, Alexa.

Alexa memiliki fitur unggulan yang tidak dimiliki oleh *voice control system* seperti yang lainnya, yaitu *Alexa Skills Kit*. Dengan fitur tersebut, setiap pengembang dapat menambahkan kemampuan alexa untuk menerjemahkan perintah-perintah baru khususnya di bidang *home automation* dimana perangkat Echo dapat digunakan sebagai jembatan antara *smart home* dengan perangkat-perangkat elektronik. *Smart home* juga membutuhkan sebuah sistem keamanan yang dapat dikontrol untuk mendeteksi adanya pergerakan yang mencurigakan didalam rumah dengan menggunakan sensor-sensor yang dapat mendeteksi suatu pergerakan ataupun getaran. Sensor ini sangat berguna untuk mendeteksi keadaan rumah yang kosong. Badan Pusat Statistik (2017) mengungkapkan bahwa persentase rumah tangga yang mengalami kejahatan pencurian pada setiap provinsi pada tahun 2011, 2012 dan 2013 berkisar antara 63,74–89,42 persen, 54,63–88,60 persen dan 66,58–90,51 persen. Dari data tersebut dapat disimpulkan bahwa persentase korban pencurian cenderung meningkat tiap tahunnya (Badan Pusat Statistik, 2014).

Berdasarkan latar belakang tersebut, peneliti bermaksud untuk mengimplementasikan Alexa pada Raspberry Pi, nantinya perangkat tersebut akan menerima perintah suara yang selanjutnya diterjemahkan untuk melakukan pendeteksian melalui sensor yang dapat mendeteksi pergerakan dan getaran seperti sensor *PIR (Passive Infra Red)*, *Microwave*, *Vibration* dan *Ultrasonic* yang di implementasikan di beberapa ruangan dan sudut rumah. Diharapkan penelitian ini mampu memberikan alternatif pada salah satu perangkat *smart home* yaitu Sistem keamanan rumah pintar berbasis *Alexa voice command* pada Raspberry Pi.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dikemukakan, rumusan masalah pada penelitian ini antara lain :

1. Bagaimana mengimplementasikan *Alexa voice command* pada Raspberry Pi ?
2. Bagaimana Raspberry Pi dapat menerjemahkan perintah yang dikirim melalui *Alexa voice command* ?
3. Bagaimana melakukan *monitoring* keadaan rumah dengan menggunakan sensor *vibration*, *microwave*, *PIR*, dan ultrasonik melalui *Alexa voice command* ?

## 1.3 Tujuan

Berikut merupakan tujuan penelitian, antara lain :

1. Dapat mengimplementasikan *Alexa voice command* pada Raspberry Pi
2. Dapat menerjemahkan perintah yang dikirim melalui *Alexa voice command*
3. melakukan *monitoring* keadaan rumah dengan menggunakan sensor *vibration*, *microwave*, *PIR*, dan ultrasonik melalui *Alexa voice command*

## 1.4 Manfaat

Memberikan gambaran implementasi alexa pada Raspberry Pi sehingga dapat memudahkan setiap orang dalam melakukan kontrol *home security device* pada *smart home* berbasis *Alexa voice command*.

## 1.5 Batasan masalah

Agar pembahasan dalam penelitian ini dapat dilakukan secara terarah dan mendapatkan hasil sesuai dengan yang diharapkan, maka perlu diterapkan batasan permasalahan. Batasan-batasan permasalahan antara lain :

1. Perintah suara menggunakan Bahasa Inggris.
2. Pengucapan perintah suara harus sesuai dengan kalimat yang telah ditentukan di *Alexa Skills Kit*.
3. Pengiriman data maksimal 16 meter.
4. Sistem membutuhkan koneksi Internet.

## 1.6 Sistematika pembahasan

Uraian singkat mengenai metodologi penelitian pada masing-masing bab adalah sebagai berikut :

### BAB I : Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan dari "Home Security Berbasis Voice Command Pada Raspberry Pi".

### BAB II : Landasan Kepustakaan

Pada bab ini akan menjelaskan tentang landasan teori yang terkait dengan penelitian. Pada bab ini juga dijelaskan tentang penelitian serupa yang pernah dilakukan.

**BAB III : Metode Penelitian**

Membahas tentang langkah kerja yang dilakukan dalam penulisan diantaranya studi literatur, analisis kebutuhan sistem, desain sistem, dan Implementasi Alexa pada Raspberry Pi.

**BAB IV : Rekayasa Kebutuhan**

Bab ini menjelaskan perancangan sistem penelitian serta implementasi sistem baik dari internal sistem itu sendiri hingga eksternal sistem.

**BAB V : Perancangan dan Implementasi**

Membahas pengujian dan analisis berdasarkan perancangan dan implementasi. Pengujian ini dilakukan dengan menggunakan parameter keberhasilan perintah suara dan tingkat akurasi.

**BAB VI : Pengujian dan Analisis**

Membuat kesimpulan yang diperoleh dari hasil pengujian dan analisis berdasarkan perancangan dan implementasi serta saran-saran untuk pengembangan lebih lanjut.

**BAB VII : Penutup**

Bab ini berisi kesimpulan atas penelitian yang telah dilakukan, dan memberikan saran untuk pengembangan sistem lebih lanjut

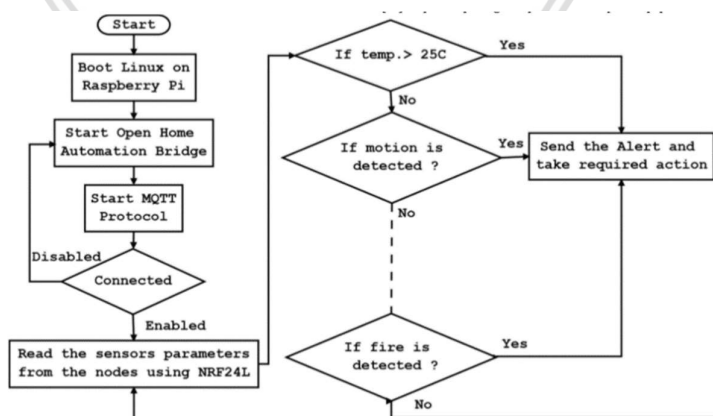


## BAB 2 LANDASAN KEPUSTAKAAN

Penelitian ini mengkaji beberapa jurnal penelitian terkait yang telah dipublikasikan diantaranya *"Scheduler and Voice Recognition on Home Automation Control System"* oleh Syarif dan Syahrial (2015) yang memanfaatkan Raspberry Pi sebagai gateway pengontrol peralatan-peralatan. Chan Zhen Yue dan Shum Ping (2017) membahas tentang design dan implementasi sistem *smart home* menggunakan mikroprosesor yang murah dan dapat disesuaikan dengan kebutuhan pengguna. Selanjutnya penelitian yang berjudul *"Eyrle Smart Home Automation using Internet of Things"* oleh Ayush Panwar dan Anandita Singh (2017) yang memanfaatkan Raspberry Pi sebagai gateway pengontrol berbagai sensor. Dan yang terakhir adalah *"Internet of Things using Node-Red and Alexa"* oleh Anoja Rajalakshmi dan Hamid Shahnasser (2017) dimana penelitian mereka berfokus pada penggunaan MQTT (Message Queue Telemetry Transport) sebagai media komunikasi *machine-to-machine* (M2M).

### 2.1 Tinjauan Pustaka

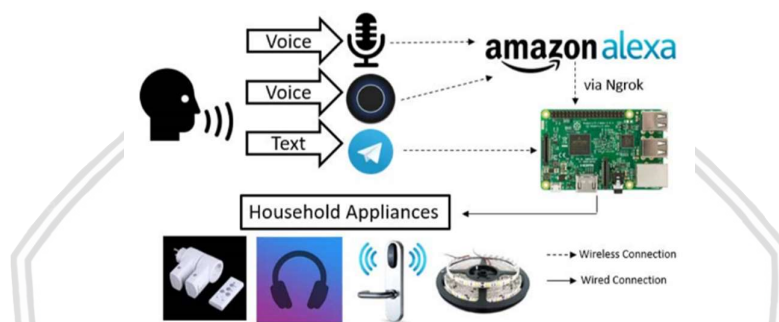
Penelitian yang sedang berkembang saat ini pada otomasi rumah berfokus pada otomasi rumah berbasis IoT. Dari penelitian tersebut menghasilkan beberapa pertimbangan baik dari segi konsumsi daya, jarak pengoperasian, kenyamanan dan biaya keseluruhan sistem. Ayush Panwar dkk. (2017) mengajukan sebuah solusi yaitu Sistem Otomasi Rumah Pintar berbasis *Internet of Things* dimana Raspberry Pi digunakan sebagai Hub pusat dari beberapa sensor yang terhubung ke Arduino Nano. Flowchart sistem yang mereka ajukan dapat dilihat pada Gambar 2.1. Mereka menggunakan *Mosquitto Broker* pada MQTT untuk menerima dan mengirim pesan yang memungkinkan pertukaran berbagai informasi dari program yang berbeda menggunakan jalur yang menyediakan layanan *publish-subscribe* (Ayush Panwar dkk., 2017).



Gambar 2. 1 Flowchart Sistem Otomasi Rumah Pintar berbasis IoT

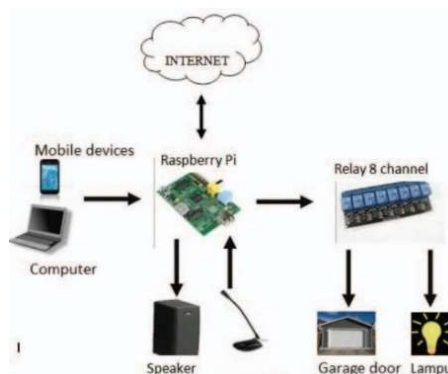
Supaya sistem tersebut dapat bekerja, modul NRF24L diperlukan untuk mengirim dan menerima data dari Hub ataupun menuju Hub. Bagaimanapun juga sistem mereka hanya bekerja jika pengguna menyalakan alat tersebut secara manual, demikian, membuat sistem tersebut kurang praktis.

Chan Zhen Yue dkk. (2017) pada penelitiannya menggunakan Ngrok Web Service untuk melakukan komunikasi dengan Raspberry Pi dan menggunakan aplikasi Telegram pada smartphone sebagai input teks. Setelah Raspberry Pi menerima perintah suara atau teks, perintah tersebut akan dikirimkan ke database pada server untuk di kenali, lalu perintah tersebut akan dikembalikan ke perangkat terkait melalui Ngrok (Chan Zhen Yue et al., 2017). Struktur sistem mereka dapat dilihat pada Gambar 2.2.



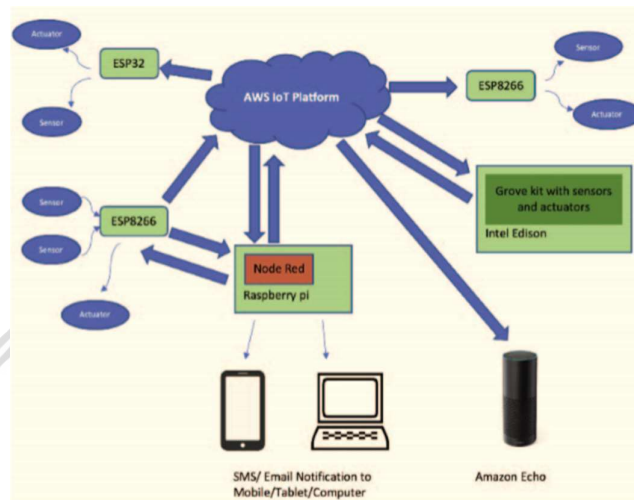
Gambar 2. 2 Struktur Voice Control System Berbasis iOS

Syarif dan Syahrial (2015) menggunakan perangkat Raspberry Pi sebagai pusat pengendali otomasi rumah. Otomasi rumah yang diajukan memiliki 3 cara berbeda untuk mengontrol peralatan elektronik. Pertama secara manual dengan tombol, kedua melalui *web interface* dan menggunakan *voice command*. *Voice command* pada sistem tersebut hanya digunakan untuk sekedar mengontrol keadaan dari peralatan elektronik yaitu kondisi on atau off, tidak sampai mengontrol suatu kondisi tertentu contohnya mengatur intensitas cahaya lampu atau suhu dari *airconditioner* (Syarif dan Syahrial, 2015). Arsitektur dari sistem mereka dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Arsitektur Scheduler and Voice Recognition on Home

Penelitian selanjutnya yaitu pegintegrasian Node-Red untuk memberikan kemudahan dalam konektivitas dengan hanya menggambar kabel atau jalur dan selanjutnya menambahkan parameternya. Node-Red yang terhubung dengan Raspberry Pi, menggunakan MQTT untuk berkomunikasi dengan perangkat *PIR*. Anoja Rajalakshmi dkk. (2017) menggunakan Intel Edison untuk menghubungkan beberapa input sensor atau actuator ke dalam *cloud*. Blok diagram Node-red dan alexa dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Blok Diagram IoT menggunakan Node-Red dan Alexa

## 2.2 Dasar Teori

Bab ini membahas beberapa teori terkait Sistem Keamanan Rumah Pintar Berbasis *Alexa Voice Command* pada Raspberry Pi. Beberapa diantaranya yaitu *Smart Home*, *Speech Recognition*, *Alexa Voice Command* dan *Raspberry Pi*.

### 2.2.1 Smart Home

*Smart Home* atau biasa dikenal dengan rumah otomatis, bangunan pintar, atau sistem rumah terintegrasi adalah sebuah pengembangan desain rumah masa kini. *Smart home* menghubungkan berbagai perangkat yang ada di dalam rumah. Dahulu, konsep awal teknologi rumah pintar hanya digunakan untuk melakukan kontrol terhadap lingkungan sekitarnya seperti cahaya dan suhu, tetapi pada saat ini penggunaan dari teknologi cerdas telah berkembang sehingga hampir semua peralatan elektronik yang terdapat di rumah dapat di kontrol menggunakan teknologi berbasis *smart home* (Vincent dkk., 2006).

Selebihnya lagi, teknologi rumah pintar tidak hanya digunakan untuk mematikan dan menyalakan peralatan, teknologi tersebut dapat memonitor lingkungan internal dan aktifitas yang terjadi di rumah. Seiring berkembangnya teknologi, saat ini untuk melakukan kontrol perangkat-perangkat yang terdapat di rumah pintar dapat dilakukan melalui *smartphone*, *website*, bahkan melalui perintah suara.

### 2.2.2 Speech Recognition

*Speech recognition* adalah proses identifikasi yang dilakukan komputer untuk mengenali kata yang diucapkan oleh seseorang tanpa mempedulikan identitas orang terkait dengan melakukan konversi sebuah sinyal akustik yang ditangkap oleh perangkat input suara (Anusuya dkk., 2009). Pola kerja *speech recognition* adalah mencocokkan sinyal akustik yang diterima dengan data yang tersimpan dalam *template* ataupun *database*. Proses pencocokan memiliki dua model utama yaitu Model Akustik yang terdiri dari fonem yang memiliki nilai tertentu yang diambil dari sinyal akustik dan Model bahasa berupa metode yang mengestimasi satu kata diikuti oleh serangkaian kata lainnya.

*Speech Recognition* juga merupakan sistem yang digunakan untuk mengenali perintah kata dari suara manusia dan kemudian diterjemahkan menjadi suatu data yang dimengerti oleh komputer. Pada saat ini, sistem ini digunakan untuk menggantikan peranan input dari keyboard dan mouse. Keuntungan dari sistem ini adalah pada kecepatan dan kemudahan dalam penggunaannya. Kata – kata yang ditangkap dan dikenali bisa jadi sebagai hasil akhir, untuk sebuah aplikasi seperti *command & kontrol*, penginputan data, dan persiapan dokumen. Parameter yang dibandingkan ialah tingkat penekanan suara yang kemudian akan dicocokkan dengan *template database* yang tersedia. Sedangkan sistem pengenalan suara berdasarkan orang yang berbicara dinamakan *speaker recognition*.

### 2.2.3 Alexa Voice Command

Alexa merupakan sebuah *Intelligent Personal Assistant* yang dikembangkan oleh Amazon dan populer berkat Amazon Echo dan perangkat Amazon Echo Dot yang dikembangkan oleh Amazon Lab126. Alexa pertama kali dikenalkan Amazon pada November 2014 bersama dengan Echo, alexa terinspirasi dari *computer voice* dan sistem percakapan pada film fiksi sains Stark Trek. Pada Januari 2017, konferensi alexa pertama diadakan di Nashville yang mengumpulkan komunitas pengembang Alexa. Hingga saat ini alexa telah dikembangkan hingga ke ranah *home automation* (Grant Clauser, 2017).

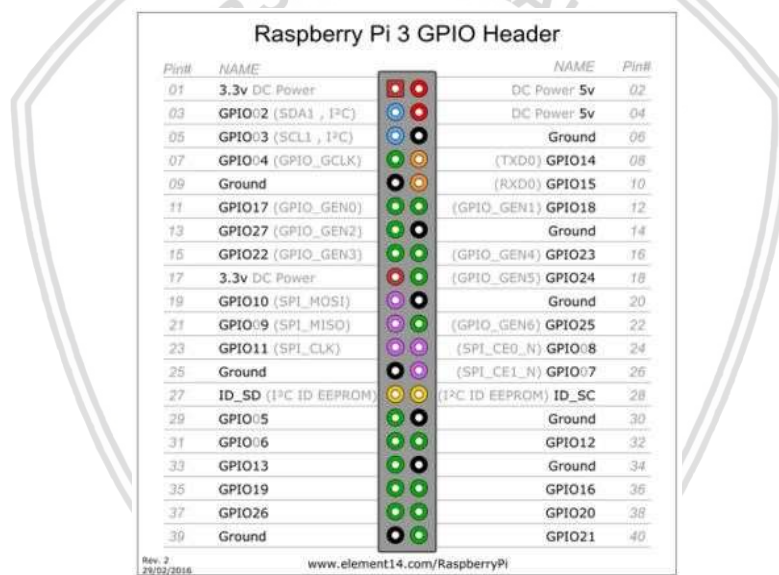
Salah satu fitur alexa yang dapat digunakan pihak pengembang alexa adalah *Alexa Skills Kit*. Fitur tersebut memungkinkan para pengembang untuk membuat dan mem-*publish* kemampuan untuk alexa menggunakan *Alexa Skills Kit*. Dengan fitur tersebut, setiap pengembang dapat menambahkan kemampuan alexa untuk menerjemahkan perintah-perintah baru khususnya di bidang *home automation*.

### 2.2.4 Raspberry Pi

Raspberry Pi adalah sebuah komputer papan tunggal (*single-board computer*) atau SBC berukuran kartu kredit. Raspberry Pi telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (*System-on-chip*) ARM yang dikemas dan diintegrasikan diatas PCB. Perangkat ini menggunakan kartu SD untuk booting dan penyimpanan jangka panjang. Desain Raspberry Pi didasarkan seputar SoC (System-on-a-chip) Broadcom BCM2835, yang telah

menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 Megabyte RAM (model B). Penyimpanan data didisain tidak untuk menggunakan hard disk atau solid-state drive, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang.

Raspberry Pi bersifat open source (berbasis Linux), Raspberry Pi bisa dimodifikasi sesuai kebutuhan penggunaannya. Sistem operasi utama Raspberry Pi menggunakan Debian GNU/Linux dan bahasa pemrograman Python. Salah satu pengembang OS untuk Raspberry Pi telah meluncurkan sistem operasi yang dinamai Raspbian, Raspbian diklaim mampu memaksimalkan perangkat Raspberry Pi. Salah satu fitur powerful dari Raspberry Pi adalah sederetan GPIO (general purpose input/output) Pinnya. Pada Raspberry Pi 3 terdapat 40 pin, setiap pin-pin tersebut mempunyai fungsinya masing-masing. Input pada pin GPIO bukan hanya sekedar dari switch melainkan dapat dari sensor atau sinyal dari komputer lainnya, begitu juga dengan output yang dapat melakukan banyak hal. Gambar 2.5 di bawah merupakan gambar GPIO Header dari Raspberry Pi 3.



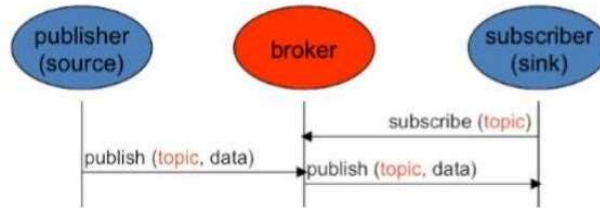
Gambar 2. 5 Raspberry Pi 3 Model B GPIO 40 Pin Block Pinout

## 2.2.5 MQTT

*Message Queuing Telemetry Transport* (MQTT) adalah sebuah protokol konektivitas *machine to machine* (M2M) atau *Internet of Things* (IoT) yang berbasis *open source* (Eclipse) dengan standard terbuka (Oasis). Protokol MQTT menggunakan prinsip *publish-subscribe*. *Publisher* menghasilkan informasi tertentu dan menerbitkan informasi tersebut, sedangkan *subscriber* merupakan client yang tertarik untuk mendapatkan informasi tertentu dengan mendaftarkan minat dari informasi tersebut. Dan terdapat “broker” yang menjamin *subscriber*



mendapatkan informasi yang disajikan oleh *publisher* (Weiss dkk., 2013). Mekanisme komunikasi *publish-subscribe* dapat dilihat pada Gambar 2.6



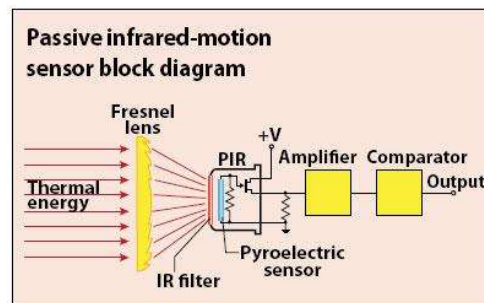
Gambar 2. 6 Mekanisme Komunikasi *Publish/Subscribe*

### 2.2.6 Sensor PIR

Sensor PIR (Passive Infra Red) adalah sensor yang digunakan untuk mendeteksi adanya pancaran sinar infra merah. Sensor PIR bersifat pasif, artinya sensor ini tidak memancarkan sinar infra merah tetapi hanya menerima radiasi sinar infra merah dari luar. Sensor ini biasanya digunakan dalam perancangan detektor gerakan berbasis PIR. Block Diagram dari sensor PIR dapat dilihat pada Gambar 2.8. Keadaan ruangan dengan perubahan temperatur pada manusia dalam suatu ruangan menjadi nilai awal (set point) yang menjadi acuan dalam sistem pengontrolan. Perubahan temperatur pada manusia dalam ruangan akan terdeteksi oleh Sensor PIR. Dikatakan PIR (Passive Infrared Receiver) karena sensor ini hanya mengenali lingkungan tanpa adanya energi yang harus dipancarkan. PIR merupakan kombinasi sebuah kristal pyroelectric, filter dan lensa Fresnel (Ayudilah, 2000).



Gambar 2. 7 Passive Infra Red (PIR)

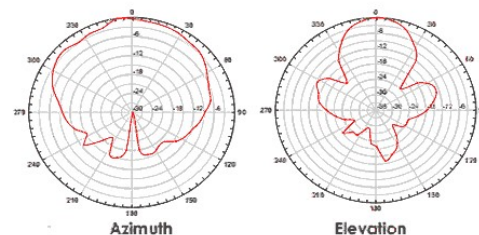


Gambar 2. 8 Block Diagram Sensor PIR

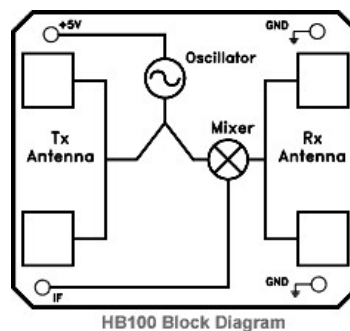
### 2.2.7 Sensor Microwave

Sensor *Microwave* adalah perangkat elektronik yang mendeteksi gerakan dan dapat digunakan untuk mengontrol lumener. Cara kerja sensor ini berbeda dengan sensor PIR, sensor ini bekerja dengan cara memproyeksikan gelombang mikro yang memantul permukaan dan kembali ke sensor di dalam detektor. Dengan menganalisis informasi ini, sensor dapat mendeteksi gerakan apa pun dalam jangkauannya dan melakukan semua ini dalam waktu kurang dari satu mikrodetik. Sensor microwave digunakan dalam lingkungan di mana panas tubuh tidak dapat dideteksi, seperti area dingin di mana orang mungkin mengenakan lapisan termal atau gudang menggunakan alat pengangkat otomatis.

Daya pancar gelombang efektif (EIRP, Effective Isotropic Radiated Power) tipikal dari modul ini sebesar  $15 \pm 20\%$  dBm, dirancang untuk memenuhi persyaratan Federal Communications Commission, FCC rules Part 15 Section 15.245 (spesifikasi penggunaan alat elektronika untuk penggunaan dalam bangunan dan/atau untuk membuka pintu bangunan/ruangan), dengan tingkat emisi semu (spurious emission) kurang dari -7,3 dBm. Kekuatan signal pantulan yang diterima (RSS, Received Signal Strength) berkisar sekitar 200  $\mu$ VP-P (diukur pada tingkat degradasi signal bolak-balik sebesar 93dB) dengan tingkat derau (noise rating) kurang dari 5  $\mu$ VRMS, diukur pada terminal keluaran rentang frekuensi 10 - 100 Hz dalam ruang Anechoic. Pada lebar pita antena 3 dB, gelombang dipancarkan secara 3 dimensi pada sudut azimuth  $80^\circ$ , elevasi  $40^\circ$ . Diagram gelombang yang dipancarkan sensor Microwave dapat dilihat pada Gambar 2.9.



Gambar 2. 9 Pola sinyal yang dipancarkan oleh sensor Microwave

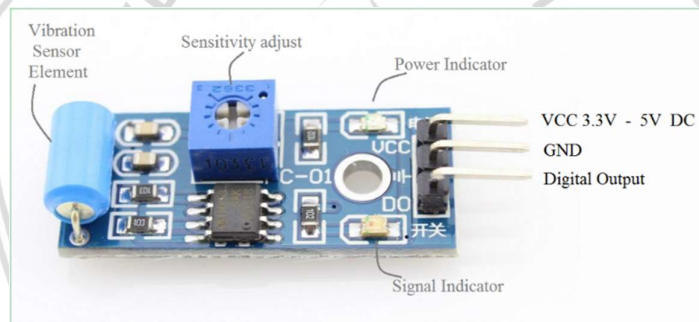


Gambar 2. 10 Block Diagram Sensor Microwave



### 2.2.8 Sensor Vibration

Modul sensor digital ini akan menghasilkan keluaran logika HIGH pada saat mendeteksi vibrasi / getaran, dapat diaplikasikan pada sistem keamanan, deteksi gempa, pendeteksi malfungsi pada sistem mekanik, analisa struktur konstruksi berdasarkan vibrasi, pengukuran kekuatan tumbukan secara tidak langsung, dsb. Inti dari modul ini adalah komponen pendeteksi getaran SW-420 yang bereaksi terhadap getaran dari berbagai sudut. Pada kondisi statis / tanpa getaran, komponen elektronika ini berfungsi seperti saklar yang berada pada kondisi menutup (normally closed) dan bersifat konduktif, sebaliknya pada kondisi terguncang (terpapar getaran) saklar akan membuka / menutup dengan kecepatan pengalihan (switching frequency) proporsional dengan kekerapan guncangan. Pengalihan bergantian secara cepat ini mirip seperti cara kerja PWM (pulse width modulation) yang merupakan sinyal pseduo-analog berupa tingkat tegangan yang kemudian dibandingkan oleh sirkuit terpadu LM393 (Voltage Comparator IC) dengan besar nilai ambang batas (treshold) tegangan pembanding diatur oleh sebuah resistor eksternal (Julian, 2015). Pin-pin dari sensor Vibration SW-420 dapat dilihat pada Gambar 2.11.



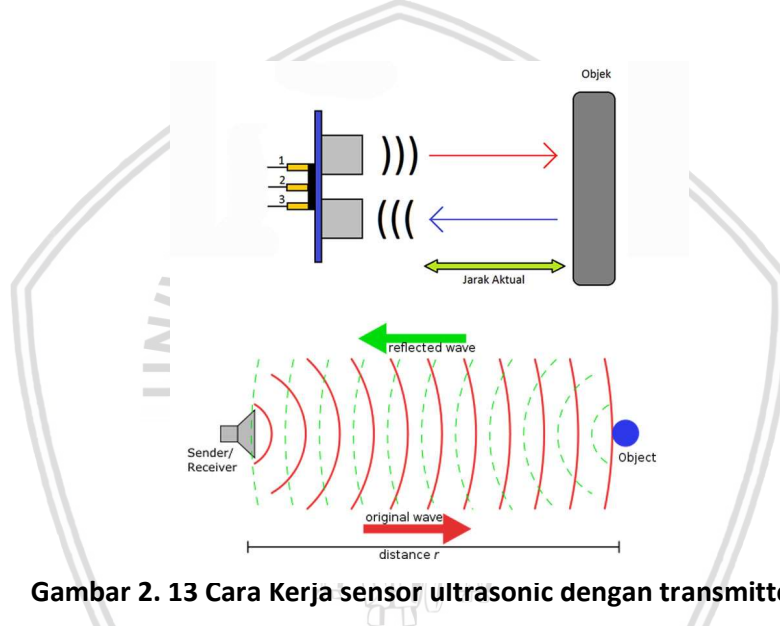
Gambar 2. 11 Sensor Vibration SW-420

### 2.2.9 Sensor Ultrasonic

Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat di dengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik bisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa (Heri Santoso, 2015). Cara kerja Sensor Ultrasonik dapat dilihat pada Gambar 2.13.



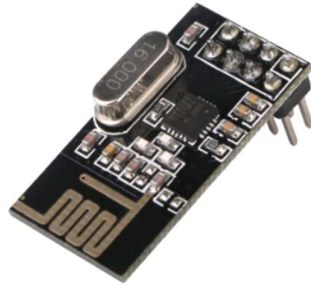
**Gambar 2. 12 Sensor Ultrasonik**



**Gambar 2. 13 Cara Kerja sensor ultrasonic dengan transmitter**

#### **2.2.10 Modul nRF24L01**

NRF24L01 merupakan modul komunikasi jarak jauh yang menggunakan frekuensi pita gelombang radio 2.4-2.5 GHz ISM (Industrial Scientific and Medical). nRF24L01 memiliki kecepatan sampai 2Mbps dengan pilihan opsi data rate 250 Kbps, 1 Mbps, dan 2 Mbps. Output daya, saluran frekuensi, dan setup protokol yang mudah diprogram melalui antarmuka SPI. Konsumsi arus yang digunakan sangat rendah, hanya 9.0mA pada daya output -6dBm dan 12.3mA dalam mode RX. Built-in Power Down dan mode standby membuat penghematan daya dengan mudah realisasi (Nordic Semiconductor ASA., 2006). Modul Komunikasi nRF24L01 dapat dilihat pada Gambar 2.14 berikut.



**Gambar 2. 14 Modul Komunikasi nRF24L01**

### **2.2.11 Mikrokontroler Arduino**

Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri (Heri Santoso, 2017). Saat ini Arduino sangat populer di seluruh dunia. Bahasa yang dipakai dalam Arduino bukan assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (libraries) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler. Bentuk dari Mikrokontroler Arduino Uno dapat dilihat pada Gambar 2.15 berikut.



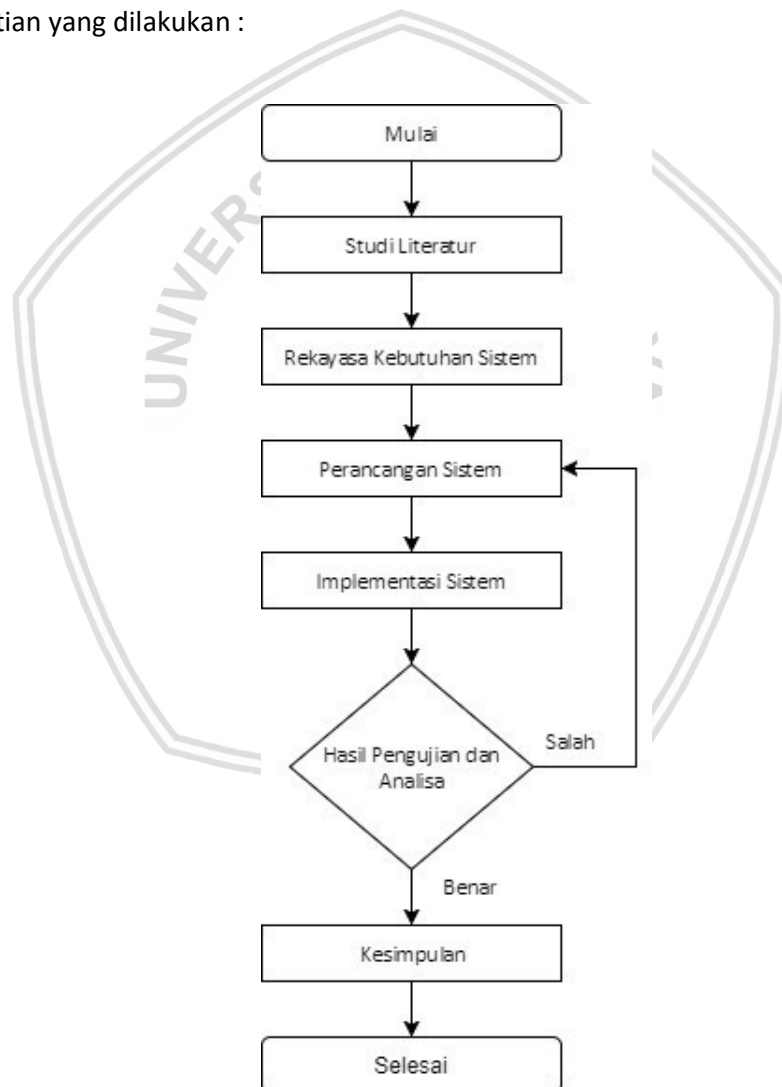
**Gambar 2. 15 Mikrokontroler Arduino Uno**

## BAB 3 METODOLOGI

Pada Bab 3 Metodologi ini akan menjelaskan bagaimana pengimplementasian sistem yang terdiri dari alur metodologi penelitian, studi literatur, rekayasa kebutuhan sistem, perancangan sistem, pengujian dan analisa serta penutup.

### 3.1 Metode Penelitian

Metode Penelitian pada Tugas Akhir ini merupakan langkah-langkah pengerjaan dan cara penyelesaian Tugas Akhir, berikut ini merupakan alur metode penelitian yang dilakukan :



Gambar 3. 1 Metodologi Penelitian

### 3.2 Studi Literatur

Pada Penelitian tugas akhir ini, studi literatur dilakukan untuk pemahaman terhadap tinjauan pustaka dan dasar teori, berikut ini merupakan studi literatur yang dilakukan:

#### 1. *Smart Home*

Studi literatur mengenai *Smart Home* dilakukan dengan memahami buku dan paper terkait *Smart Home*. Bagaimana mengimplementasikan *smart home*. Pada tahap ini, peneliti mempelajari apa saja komponen yang terdapat pada *smart home*

#### 2. *Speech Recognition*

Studi literatur mengenai *Speech Recognition* dilakukan kajian dari beberapa literatur terkait baik dari jurnal, buku maupun publikasi di internet. Penelitian ini mempelajari bagaimana cara *Speech Recognition* bekerja.

#### 3. *Alexa Voice Command*

Pada studi literatur mengenai *Alexa Voice Command*, peneliti mempelajari bagaimana cara mengimplementasikan *alexa voice command* pada Raspberry Pi dan mempelajari bagaimana menambahkan fitur atau kemampuan dari alexa untuk dapat menerjemahkan perintah yang di kirim melalui perangkat Raspberry Pi

#### 4. MQTT

Pada studi literatur mengenai MQTT, peneliti akan melakukan kajian pada jurnal yang menjelaskan mekanisme sistem *publish-subscribe* yang merupakan cara komunikasi pesan secara tidak langsung atau indirect communication pada suatu jaringan. Pada penelitian ini dipelajari bagaimana cara kerja sistem *publish-subscribe* secara umum dan protokol yang bekerja dalam mendukung sistem tersebut

#### 5. Sensor PIR (*Passive Infra Red*)

Pada studi literatur mengenai Sensor PIR, peneliti akan melakukan kajian pada jurnal yang menjelaskan mekanisme sistem kerja Sensor PIR. Pada penelitian ini dipelajari bagaimana cara kerja Sensor PIR secara umum dan cara menghubungkannya pada *Arduino Uno Board* dalam mendukung sistem tersebut.

#### 6. Sensor *Microwave*

Pada studi literatur mengenai Sensor *Microwave*, peneliti akan melakukan kajian pada jurnal yang menjelaskan mekanisme sistem kerja Sensor *Microwave*. Pada penelitian ini dipelajari bagaimana cara kerja Sensor *Microwave* secara umum dan cara menghubungkannya pada *Arduino Uno Board* dalam mendukung sistem tersebut.

#### 7. Sensor *Vibration*

Pada studi literatur mengenai Sensor *Vibration*, peneliti akan melakukan kajian pada jurnal yang menjelaskan mekanisme sistem kerja Sensor *Vibration*. Pada penelitian ini dipelajari bagaimana cara kerja Sensor *Vibration* secara umum dan cara menghubungkannya pada *Arduino Uno Board* dalam mendukung sistem tersebut.

#### 8. Sensor *Ultrasonic*

Pada studi literatur mengenai Sensor *Ultrasonic*, peneliti akan melakukan kajian pada jurnal yang menjelaskan mekanisme sistem kerja Sensor *Ultrasonic*. Pada penelitian ini dipelajari bagaimana cara kerja Sensor *Ultrasonic* secara umum dan cara menghubungkannya pada *Arduino Uno Board* dalam mendukung sistem tersebut.

#### 9. Modul nRF24L01

Pada studi literatur mengenai modul nRF24L01, peneliti akan melakukan kajian pada jurnal yang menjelaskan mekanisme sistem kerja modul nRF24L01. Pada penelitian ini dipelajari bagaimana cara kerja modul nRF24L01 secara umum dan cara menghubungkannya pada *Arduino Uno Board* dalam mendukung sistem tersebut.

#### 10. Arduino

Pada studi literatur mengenai *Arduino Board*, peneliti akan melakukan kajian pada jurnal yang menjelaskan mekanisme sistem kerja Arduino. Pada penelitian ini dipelajari bagaimana Arduino mengirimkan data yang didapatkan dari sensor secara umum dan cara sensor dapat terhubung untuk mendukung sistem tersebut.

### 3.3 Rekayasa Kebutuhan Sistem

Rekayasa kebutuhan sistem dilakukan untuk mengetahui apa saja yang dibutuhkan oleh sistem agar sistem dapat dikatakan bekerja sesuai dengan tujuan sistem. Kebutuhan dari sistem pada penelitian ini terdiri dari gambaran umum sistem yang didalamnya terdapat pembahasan perspektif sistem, ruang lingkup, karakteristik pengguna, serta asumsi dan ketergantungan sistem. Selain itu terdapat pula pembahasan kebutuhan antarmuka eksternal, kebutuhan fungsional yang nantinya akan dijelaskan secara rinci pada bab selanjutnya mengenai rekayasa kebutuhan.

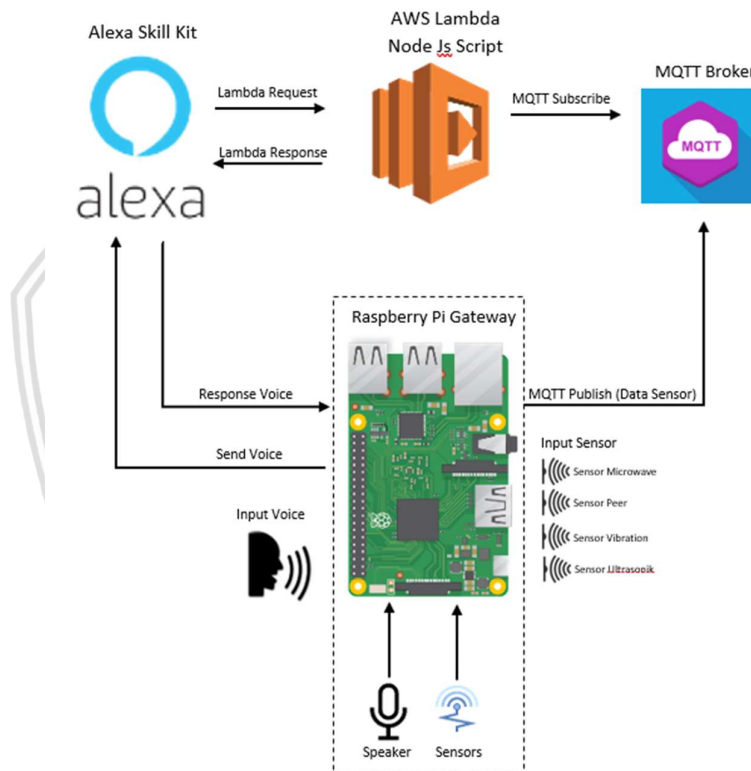
### 3.4 Perancangan Sistem

Tahap perancangan sistem ini bertujuan agar perancangan sistem penelitian yang dilakukan menjadi terstruktur. Perancangan sistem akan dimulai dengan merancang masing-masing sub sistem yang diperlukan diantaranya sub-sistem mikrofon, subsistem Sensor. Selain perancangan sistem itu sendiri terdapat pula perancangan eksternal sistem dari Amazon Web Service yaitu perancangan Alexa Skills Kit, Alexa Voice Service dan Amazon Lambda yang nantinya akan di jelaskan secara rinci pada bab mengenai perancangan sistem.



### 3.5 Implementasi Sistem

Tahap implementasi sistem ini akan menjelaskan arsitektur sistem dan bagaimana alur perintah suara tersebut di proses. Implementasi sistem ini dilakukan ketika semua proses perancangan sistem telah dipenuhi. Pembahasan tentang implementasi sistem disini antara lain menjelaskan tentang implementasi pada Raspberry Pi sebagai perangkat penerima perintah suara dan kode program untuk menerima hasil terjemahan perintah suara selain itu implementasi pada Amazon Web Service yaitu Alexa Skills Kit, Alexa Voice Service dan Amazon Lambda juga akan dibahas pada bab ini. Gambar 3.2 menunjukan arsitektur dan alur kerja sistem.



Gambar 3. 2 Arsitektur dan Alur Sistem

### 3.6 Pengujian dan Analisis

Pada tahap pengujian sistem ini menggunakan parameter yang disesuaikan pada perancangan sistem. Proses Pengujian dilakukan dengan beberapa tahap, yaitu menguji *Alexa Voice Command*, jarak pengiriman data antar node, dan keseluruhan Sistem. Tahap ini akan menjelaskan bagaimana penelitian dilakukan dari perancangan dan implementasi lalu dilakukan pengujian dan analisa yang akan dijelaskan pada bab 6.

### 3.7 Kesimpulan

Pada bab kesimpulan ini akan menjelaskan bagaimana hasil dari penelitian yang telah dilakukan dan saran yang nantinya akan membantu penelitian selanjutnya.



## BAB 4 REKAYASA KEBUTUHAN

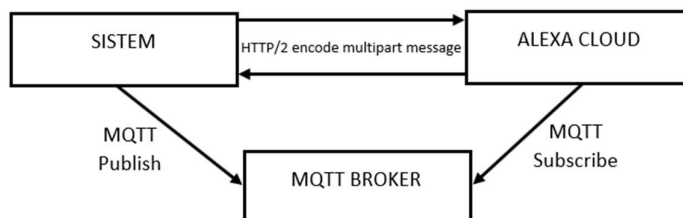
Pada Bab 4 Rekayasa Kebutuhan ini akan dijelaskan mengenai kebutuhan-kebutuhan dari sistem yang dibangun agar bisa memenuhi tujuan yang telah dibuat sebelumnya. Kebutuhan tersebut meliputi gambaran umum sistem, kebutuhan fungsional dan kebutuhan non fungsional. Dengan adanya rekayasa kebutuhan diharapkan sistem yang akan dibuat dapat bekerja dengan baik.

### 4.1 Gambaran Umum Sistem

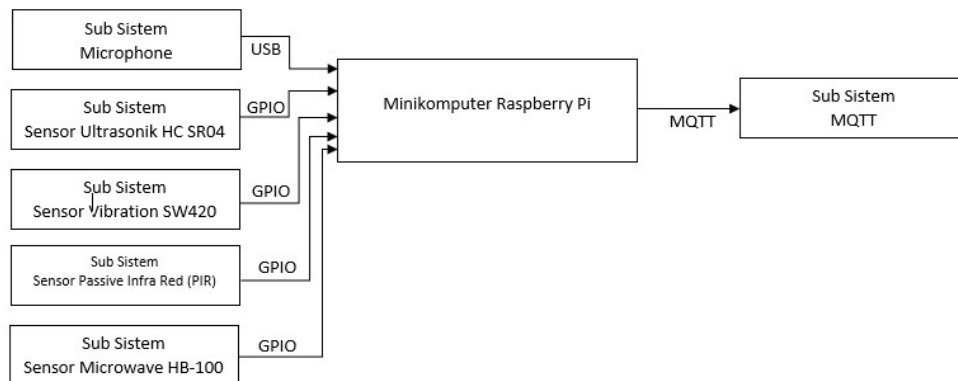
Sistem ini bekerja untuk memonitoring keadaan keamanan rumah dengan menggunakan sensor *Microwave* HB-100 sebagai sensor untuk mendeteksi gerakan, sensor ultrasonik sebagai sensor untuk mendeteksi jarak kerapatan jendela, sensor *vibration* sebagai sensor untuk mendeteksi getaran, dan sensor *PIR* sebagai sensor untuk mendeteksi gerakan yang diimplementasikan di beberapa ruangan dengan mini komputer Arduino yang dihubungkan ke NRF24L01 sebagai media pengiriman data ke Raspberry Pi menggunakan *voice command*. Perangkat Raspberry Pi berfungsi sebagai modul *speech recognition*. Raspberry tersebut nantinya akan diinstall sebuah *service* yang bernama *Alexa Voice Service* sehingga perangkat tersebut memiliki kemampuan untuk mengenali suara. Pada sistem ini, suara yang telah diterima oleh Raspberry dari sebuah *microphone* akan diteruskan ke *Amazon Web Service* dan diolah, sehingga menghasilkan teks yang nantinya digunakan sebagai pengambil keputusan. Sistem ini menggunakan metode *Publish-Subscribe* MQTT untuk melakukan pengiriman data. Data yang didapatkan dari sensor-sensor akan di *publish* ke *broker* MQTT dan di *subscribe* oleh perangkat Raspberry Pi yang kemudian dibacakan hasilnya oleh *Alexa*.

#### 4.1.1 Perspektif Sistem

Sistem ini ditujukan untuk setiap orang yang ingin mengimplementasikan teknologi *Smarthome* di dalam rumah mereka. Sistem ini akan dipasang dibagian-bagian ruangan tertentu yang strategis. Sistem ini merupakan sebuah sistem besar. Bagian sistem besar secara keseluruhan dapat dilihat pada Gambar 4.1. Didalam sistem besar tersebut, terdapat sebuah sub-sistem yang dapat dilihat pada Gambar 4.2.



Gambar 4. 1 Sistem Secara Keseluruhan



**Gambar 4. 2 Sub Sistem dari Sistem Besar**

#### 4.1.2 Ruang Lingkup

Sistem ini dibuat untuk digunakan oleh setiap orang yang menginginkan keadaan rumah yang aman dan terpantau. Pada saat ini banyak sekali metode-metode yang digunakan untuk memantau keadaan rumah mereka mulai dari cctv, alarm, sensor, dan bahkan perintah suara. Oleh karena itu sistem ini dibuat dengan memanfaatkan layanan Amazon Bernama *Alexa Voice Service* supaya sistem ini dapat mendeteksi keadaan rumah dengan menggunakan perintah suara.

#### 4.1.3 Karakteristik Pengguna

Alat ini diperuntukan bagi pengguna yang menginginkan kemudahan dalam melakukan pemantauan keadaan rumah pada sebuah *Smarthome*. Pengguna juga diharuskan memiliki kemampuan Bahasa Inggris, karena sistem ini menggunakan Bahasa Inggris sebagai perintah suaranya.

#### 4.1.4 Asumsi dan Ketergantungan

Asumsi ketergantungan pada Sistem ini antara lain :

1. Alat ini menggunakan mini USB microphone, sehingga jarak ideal pengguna pada saat berkomunikasi kepada Alexa atau memberikan perintah kepada Alexa adalah pada radius 0-4 meter dari dimana alat ditempatkan.
2. Alat ini akan mem-*publish* data sensor dari Raspberry ke *broker* MQTT dan kemudian melakukan *subscribe*, sehingga membutuhkan koneksi internet.
3. Alat ini membutuhkan sumber daya listrik dengan tegangan 5v untuk menyalakan perangkat Raspberry dan membutuhkan sumber tegangan 7-12 Volt untuk menyalakan perangkat Arduino.

## 4.2 Kebutuhan Antarmuka Eksternal

Sistem ini terdapat sebuah kebutuhan antarmuka eksternal yaitu *Amazon Web Service*. Pada *Amazon Web Service* terdapat layanan yang dipakai diantaranya *Alexa Skill Kit*, *Alexa Voice Service* dan *Amazon Lambda*. Ketiga layanan tersebut merupakan layanan yang diberikan oleh amazon dimana *Alexa Voice Service* digunakan supaya perangkat Raspberry selaku modul *speech recognition* dapat mengirimkan perintah suara yang diterima melalui microphone lalu diteruskan ke *Alexa Skill Kit* yang dimana suara tersebut akan diproses menjadi sebuah teks. Selanjutnya hasil proses tersebut dikirimkan ke *Amazon Web Service* dan dimodelkan menjadi bentuk JSON.

## 4.3 Kebutuhan Fungsional

Adapun kebutuhan fungsional yang harus dipenuhi dalam penelitian ini antara lain :

- 1. Minikomputer Raspberry dapat menjadi modul *speech recognition***  
Minikomputer Raspberry akan dipasang sebuah layanan yang bernama *Alexa Voice Service* sehingga Raspberry dapat menerima sebuah input berupa suara melalui mini USB *microphone*. Nantinya suara tersebut akan dikirimkan ke *Alexa Skill Kit*. Kemudian perintah suara tersebut akan diproses disana.
- 2. Minikomputer Raspberry dapat melakukan *Publish* dan *Subscribe* ke broker MQTT**  
Minikomputer Raspberry akan melakukan *publish* data sensor yang didapatkan ke broker MQTT. Kemudian minikomputer raspberry akan melakukan *subscribe* ke *broker* MQTT melalui perintah suara yang telah diterjemahkan dan dimodelkan didalam *AWS Lambda* untuk mendapatkan hasil data sensor.
- 3. Sistem dapat ditempatkan pada bagian-bagian tertentu secara *wireless***  
Sistem ini terdiri dari beberapa node yang telah terhubung melalui NRF24L01 sebagai media komunikasi antara pusat pengelolaan data (Perangkat Raspberry) dengan node-node (sensor dan arduino), sehingga penempatan node-node ini dapat dilakukan pada jarak yang cukup jauh dan praktis.
- 4. Sistem dapat memberitahukan keadaan rumah dengan perintah suara menggunakan Bahasa Inggris**  
Sistem akan memberitahukan keadaan rumah berdasarkan data hasil *subscribe* ke *broker* MQTT. Ketika terdapat parameter *room status* atau keadaan lokasi tertentu dari sebuah ruangan, maka sistem akan memberitahukan keadaan ruangan tersebut menggunakan suara dari Alexa, baik memberitahukan apakah ada orang diruangan tersebut atau memberitahukan ada atau tidaknya pembobolan melalui pintu ataupun jendela.
- 5. Node Sensor PIR (*Passive Infra Red*)**

Sistem ini menggunakan Sensor PIR (*Passive Infra Red*) terhubung dengan mikrokontroler Arduino yang berfungsi untuk mendeteksi gerak melalui panas tubuh manusia. Node sensor PIR dapat diletakkan pada tempat yang strategis didalam rumah yang sering dilalui oleh banyak orang misalnya pintu, meja diruang tamu, meja diruang dapur, dll.

**6. Node Sensor Microwave**

Sistem ini menggunakan Sensor Microwave yang terhubung dengan Mikrokontroler Arduino yang berfungsi untuk mendeteksi pergerakan dengan memanfaatkan gelombang radio yang dipantulkan terhadap objek. Node sensor Microwave dapat diletakkan pada tempat strategis yang sering dilalui oleh banyak orang, misal : kaki meja, pintu, lemari perhiasan dll.

**7. Node Sensor Vibration**

Sistem ini menggunakan Sesor Vibration yang terhubung dengan mikrokontroler Arduino yang berfungsi untuk mendeteksi getaran. Node Sensor Vibration akan diimplementasikan pada sebuah benda yang dapat menghasilkan getaran jika benda tersebut bergerak. Misal : pintu lemari perhiasan, daun jendela, daun pintu, laci perhiasan dll.

**8. Node Sensor Ultrasonic**

Sistem ini menggunakan Sensor Ultrasonic yang terhubung dengan mikrokontroler Arduino yang berfungsi untuk mendeteksi jarak. Node Sensor ini akan diimplementasikan pada sebuah jendela untuk mengukur jarak kerenggangan jendela.

#### 4.4 Kebutuhan Non-Fungsional

Adapun kebutuhan non-fungsional untuk memenuhi kebutuhan penelitian ini antara lain :

**1. Hardware**

**A. Personal Computer (PC) atau Laptop**

Sistem ini membutuhkan PC atau Laptop sebagai media untuk mengaktifkan *Alexa Voice Service*, sehingga minikomputer Raspberry Pi dapat menjadi modul *speech recognition*.

**B. Powerbank**

Setiap node-node yang dipasang di bagian-bagian tertentu akan membutuhkan sumber daya listrik, sehingga dibutuhkan *powerbank* sebagai sumber daya bagi node-node tersebut.

**2. Software**

**A. Arduino IDE**

Arduino IDE berfungsi sebagai perangkat lunak untuk memprogram perangkat Arduino yang digunakan pada setiap node.

**B. Node JS**

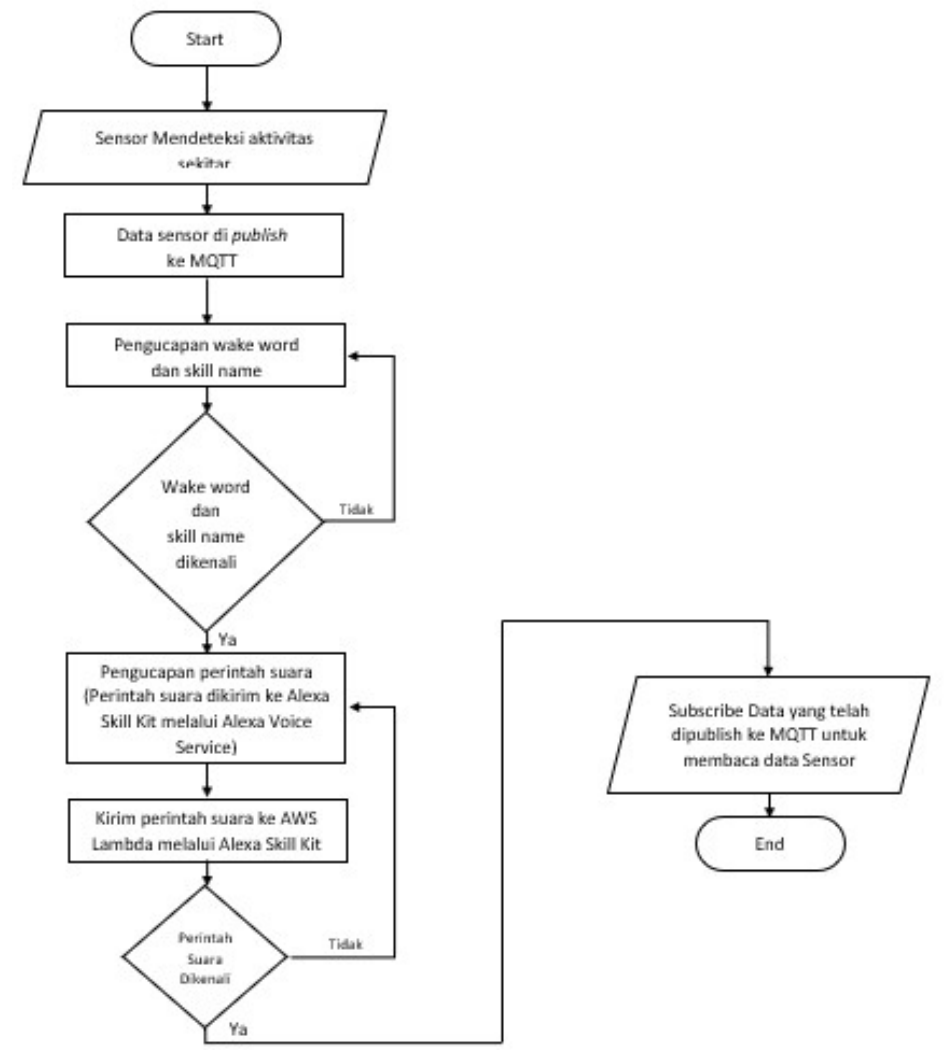
Sistem ini menggunakan Bahasa pemrograman JavaScript sehingga membutuhkan perangkat lunak yang cocok untuk Bahasa pemrograman tersebut seperti Node JS.



## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan membahas tentang perancangan dan implementasi diantaranya adalah perancangan dan implementasi dari internal sistem yaitu sub sistem mikrofon, sub sistem sensor, dan sub sistem MQTT. Selain itu dilakukan juga perancangan dan implementasi dari eksternal sistem yaitu alexa voice service, alexa skills kit dan amazon web service lambda.

Cara kerja sistem secara garis besar ditunjukkan oleh flowchart dibawah ini pada Gambar 5.1.



Gambar 5. 1 Alur Sistem Secara Keseluruhan

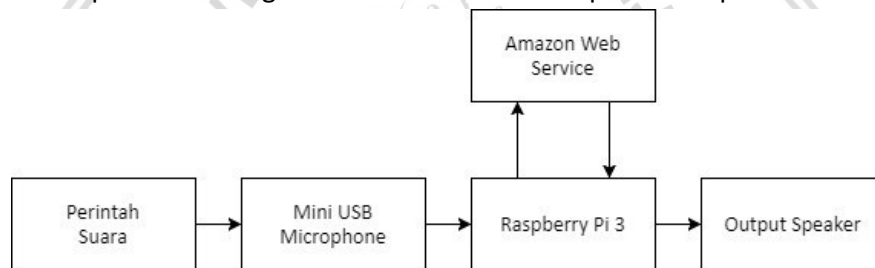
## 5.1 Perancangan

### 5.1.1 Perancangan Sub Sistem Mikrofon (Modul Speech Recognition)

Pada tahap perancangan sub sistem mikrofon (modul *speech recognition*) ditentukan penggunaan perangkat-perangkat yang dibutuhkan seperti mini komputer yang digunakan adalah raspberry pi 3 dan untuk mini mikrofonnya adalah Mini USB Microphone. Spesifikasi dari mini usb microphone tersebut adalah sebagai berikut :

1. Sensitivity: -47dB±4dB
2. Sensitivity Loss: -3dB at 1.5V
3. Work Voltage: 4.5V
4. Frequency Response: 100Hz-16kHz
5. SNR: > -67dB

Selain mini usb microphone, sub sistem ini juga akan dihubungkan dengan sebuah output speaker untuk mengeluarkan suara *response* dari perintah yang telah diucapkan. Blok diagram dari sub sistem ini dapat dilihat pada Gambar 5.2.

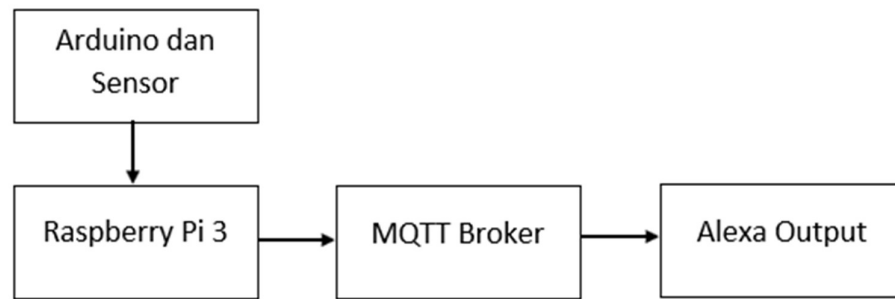


Gambar 5. 2 Blok Diagram Sub Sistem Mikrofon

### 5.1.2 Perancangan Sub Sistem Sensor PIR

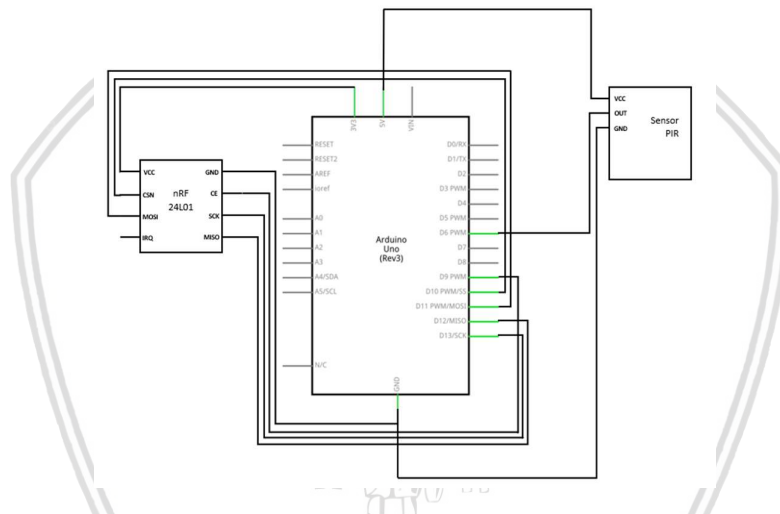
Pada tahap perancangan Sub Sistem Sensor *PIR*, akan dilakukan penentuan penempatan Pin Sensor *PIR* dan modul nRF24L01 ke pin Arduino dan penentuan penempatan nRF24L01 ke pin GPIO raspberry pi. Sub sistem ini menggunakan sensor *PIR* sebagai sensor untuk mendeteksi **gerakan** dan modul nRF24L01 sebagai media komunikasi dan transfer data dari Arduino ke Raspberry Pi. Kemudian data yang didapatkan dari sensor akan langsung *publish* ke MQTT Broker. Data-data dari sensor nantinya akan menjadi output suara yang akan diucapkan oleh Alexa. Sub Sistem ini akan diimplementasikan pada sebuah pintu utama dalam sebuah rumah untuk mendeteksi sebuah gerakan jika ada orang yang melewatinya. Selain jarak sensor yang cukup lebar dan jauh untuk mendeteksi sebuah gerakan, sensor PIR juga memiliki design hardware yang kecil dan mudah untuk ditempatkan dimana saja, sehingga Sub sensor ini sangat tepat jika ditempatkan pada sebuah pintu utama. Blok diagram sub sistem ini dapat dilihat pada Gambar 5.3.





**Gambar 5. 3 Blok diagram Sub Sistem Sensor**

Skematik koneksi antara pin Sensor *PIR* dan nRF24L01 ke pin Arduino dapat dilihat pada Gambar 5.4 berikut.



**Gambar 5. 4 Skematik Sensor *PIR* dan nRF24L01 pada Arduino Uno Board**

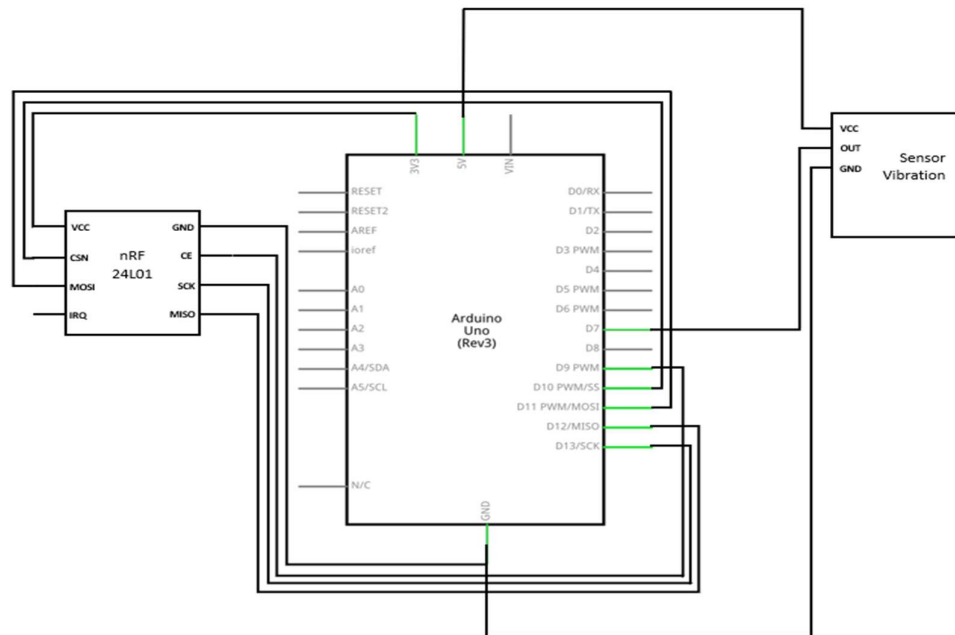
**Tabel 5. 1 Pinout antara nRF24L01 dan Sensor PIR ke Arduino Uno Board**

No	nRF24L01	Sensor PIR
1	VCC > 3,3 V	VCC > 5 V
2	CSN > D10	Output > D6
3	MOSI > D11	GND > GND
4	GND > GND	
5	CE > D9	
6	SCK > D13	
7	MISO > D12	

### 5.1.3 Perancangan Sub Sistem Sensor *Vibration*

Pada tahap perancangan Sub Sistem Sensor *Vibration*, akan dilakukan penentuan penempatan Pin Sensor *Vibration* dan modul nRF24L01 ke pin Arduino dan penentuan penempatan nRF24L01 ke pin GPIO raspberry pi. Sub sistem ini menggunakan sensor *Vibration* sebagai sensor untuk mendeteksi **getaran** dan

modul nRF24L01 sebagai media komunikasi dan transfer data dari Arduino ke Raspberry Pi. Kemudian data yang didapatkan dari sensor akan langsung *publish* ke MQTT Broker. Data-data dari sensor nantinya akan menjadi output suara yang akan diucapkan oleh Alexa. Sub sistem sensor *Vibration* akan diimplementasikan pada sebuah laci yang menyimpan barang-barang berharga agar dapat mendeteksi sebuah getaran yang dihasilkan jika laci tersebut dibuka secara paksa oleh seorang pencuri atau perampok. Sub sistem sensor ini memiliki design *hardware* yang kecil sehingga sub sistem ini sangat tepat jika ditempatkan dalam sebuah laci penyimpanan barang berharga. Skematik sub sistem ini dapat dilihat pada Gambar 5.5.



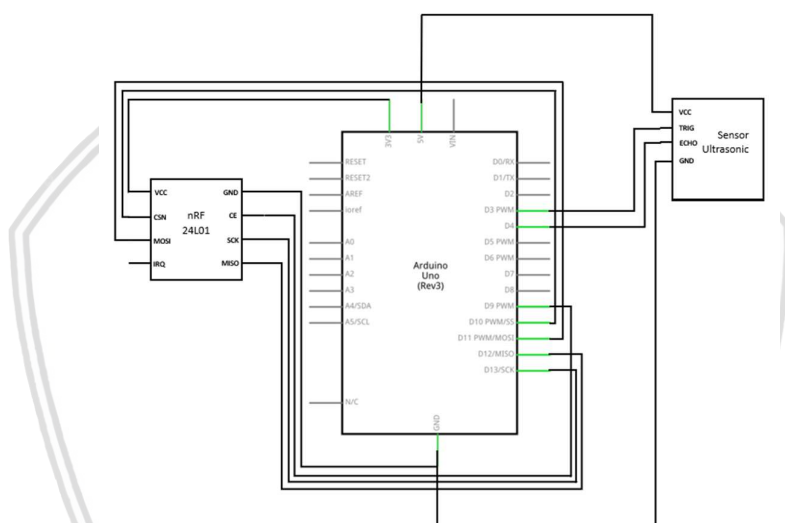
Gambar 5. 5 Skematik Sensor *Vibration* dan nRF24L01 pada *Arduino*

Tabel 5. 2 Pinout antara nRF24L01 dan Sensor *Vibration* ke *Arduino*

No	nRF24L01	Sensor <i>Vibration</i>
1	VCC > 3,3 V	VCC > 5 V
2	CSN > D10	Output > D7
3	MOSI > D11	GND > GND
4	GND > GND	
5	CE > D9	
6	SCK > D13	
7	MISO > D12	

#### 5.1.4 Perancangan Sub Sistem Sensor *Ultrasonic*

Pada tahap perancangan Sub Sistem Sensor *Ultrasonic*, akan dilakukan penentuan penempatan Pin Sensor *Ultrasonic* dan modul nRF24L01 ke pin Arduino. Sub sistem ini menggunakan sensor *Ultrasonic* sebagai sensor untuk mendeteksi jarak dan modul nRF24L01 sebagai media komunikasi untuk mentransfer data dari Arduino ke Raspberry Pi. Kemudian data yang didapatkan dari sensor *Ultrasonic* akan langsung *publish* ke MQTT Broker. Data-data dari sensor nantinya akan menjadi output suara yang akan diucapkan oleh Alexa. Sub sistem sensor ini diimplementasikan pada sebuah jendela kamar. Sehingga ketika keadaan jarak jendela dengan daun jendela ada kerenggangan lebih dari 4 cm, maka sensor microwave akan mendeteksi bahwa keadaan jendela sedang terbuka. Skematik sub sistem ini dapat dilihat pada Gambar 5.6.



Gambar 5. 6 Skematik Sensor *Ultrasonic* dan nRF24L01 pada *Arduino*

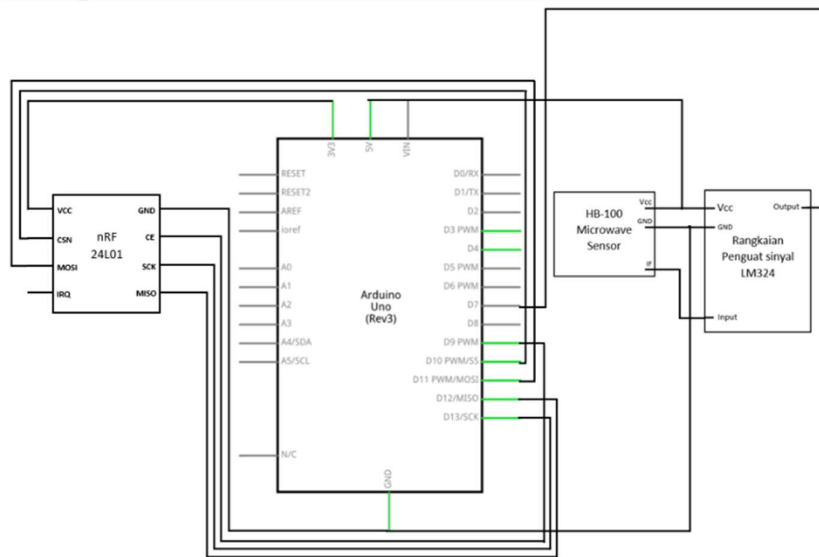
Tabel 5. 3 Pinout antara nRF24L01 dan sensor ultrasonic ke *Arduino*

No	nRF24L01	Sensor Ultrasonic
1	VCC > 3,3 V	VCC > 5 V
2	CSN > D10	TRG > D3
3	MOSI > D11	GND > GND
4	GND > GND	ECHO > D4
5	CE > D9	
6	SCK > D13	
7	MISO > D12	

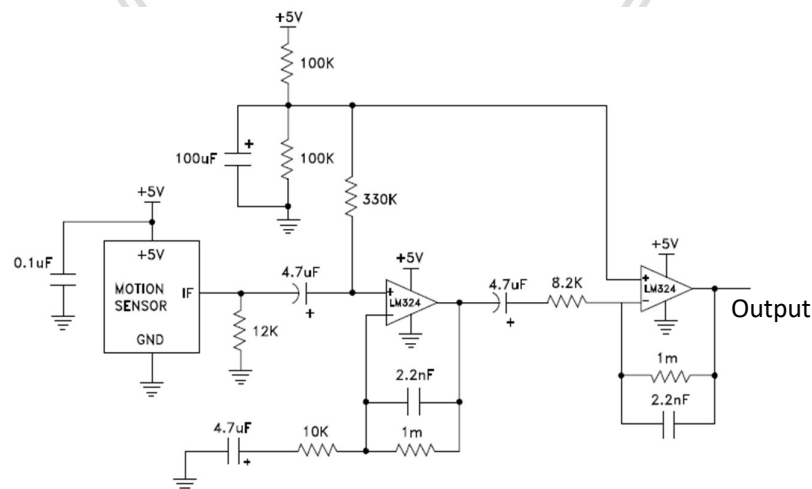
#### 5.1.5 Perancangan Sub Sistem Sensor *Microwave*

Pada tahap perancangan Sub sistem sensor *microwave*, akan dilakukan penentuan penempatan pin sensor *microwave* dan modul nRF24L01 ke pin Arduino. Sub sistem ini menggunakan sebuah rangkaian penguat sinyal dengan menggunakan modul LM324 sebagai perantara agar sensor dapat mendeteksi

gerakan. Output yang diberikan dari sub sistem ini berupa frekuensi dari sebuah pergerakan. Kemudian data yang didapatkan dari sensor *microwave* akan dikirimkan ke raspberry Pi sebagai broker MQTT dengan menggunakan modul nRF24L01 sebagai media komunikasi antara Arduino dan Raspberry Pi yang selanjutnya akan langsung di publish ke MQTT dan langsung disubscribe oleh alexa. Data yang telah di subscribe, nantinya akan diproses oleh AWS Lambda menjadi sebuah output suara. Sub sistem sensor ini dapat mendeteksi sebuah pergerakan melewati sebuah penghalang seperti tembok, kayu dan kaca. Selain itu, sub sistem ini memiliki design hardware yang cukup kecil dan sederhana, sehingga sub sistem ini dapat diimplementasikan didalam sebuah lemari penyimpanan barang berharga untuk mendeteksi sebuah pergerakan mencurigakan tanpa diketahui. Skematik sub sistem ini dapat dilihat pada gambar 5.7 dan 5.8.



Gambar 5. 7 Skematik Sensor *Microwave* dan nRF24L01 pada *Arduino*



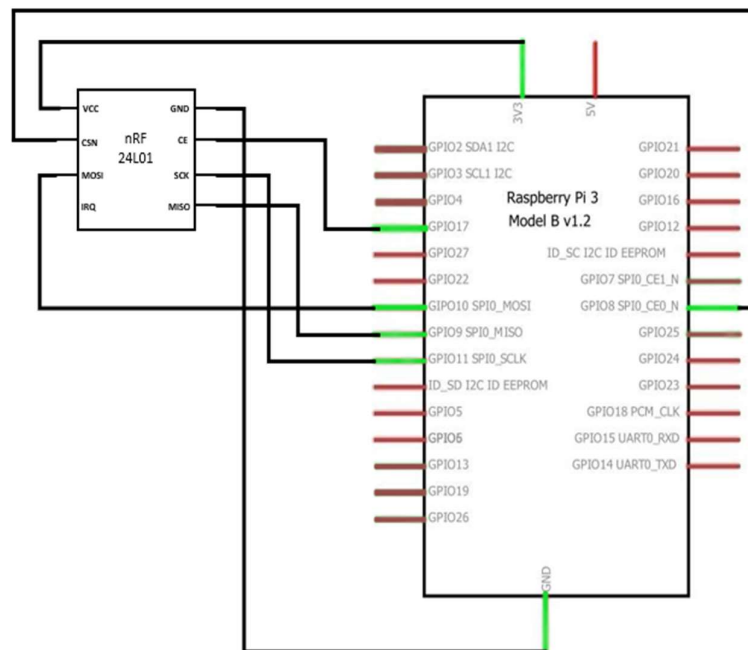
Gambar 5. 8 Skematik Rangkaian Penguat Sinyal menggunakan LM324

**Tabel 5. 4 Pinout antara nRF24L01 dan sensor Microwave**

No	nRF24L01	Sensor Microwave
1	VCC > 3,3 V	VCC > 5 V
2	CSN > D10	Output > D7
3	MOSI > D11	GND > GND
4	GND > GND	
5	CE > D9	
6	SCK > D13	
7	MISO > D12	

### 5.1.6 Perancangan Sub Sistem *Receiver*

Pada tahap perancangan Sub Sistem Receiver, akan dilakukan penentuan penempatan nRF24L01 ke pin GPIO raspberry pi. Sub sistem ini berfungsi untuk menerima semua data yang dikirimkan dari setiap node dengan menggunakan media komunikasi nRF24L01. Kemudian data yang didapatkan dari sensor akan langsung *publish* ke MQTT Broker. Data-data dari sensor nantinya akan menjadi output suara yang akan diucapkan oleh Alexa. Sub sistem ini memiliki design hardware yang sederhana dan praktis, sehingga penempatan sub sistem ini dapat dilakukan pada lokasi yang mudah untuk menerima input suara dari user seperti di ruang tamu. Skematik koneksi antara nRF24L01 ke GPIO pin Raspberry Pi sebagai *receiver*, dapat dilihat pada Gambar 5.9 berikut.



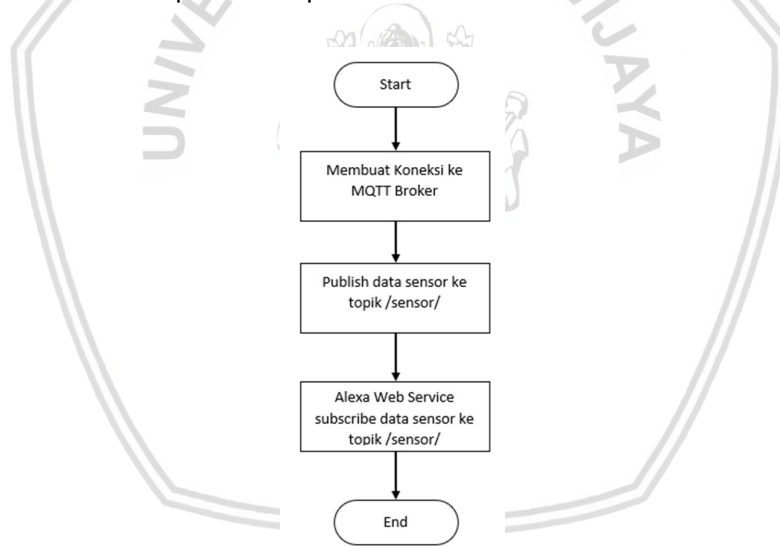
**Gambar 5. 9 Skematik nRF24L01 pada Raspberry Pi sebagai receiver**

Tabel 5. 5 Pinout antara nRF24L01 ke Raspberry Pi

No	nRF24L01
1	VCC > 3,3 V
2	CSN > GPIO8
3	MOSI > GPIO10
4	GND > GND
5	CE > GPIO17
6	SCK > GPIO11
7	MISO > GPIO9

### 5.1.7 Perancangan Sub Sistem MQTT

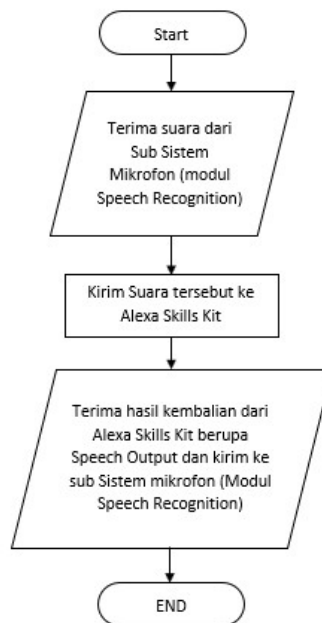
Pada tahap perancangan Sub Sistem MQTT dilakukan penentuan library MQTT yang nantinya akan dipakai. Untuk sub sistem ini library yang akan dipakai adalah MQTT.js pada Bahasa pemrograman node.js, MQTT ini nantinya akan digunakan untuk mekanisme komunikasi antara sistem dengan MQTT Broker dimana sistem akan men-*subscribe* suatu topik yang telah ditentukan yaitu */sensor/*. Alur kerja dari sub sistem ini dapat dilihat pada Gambar 5.10 berikut.



Gambar 5. 10 Alur Kerja Sub Sistem MQTT

### 5.1.8 Perancangan Alexa Voice Service

Pada tahap perancangan *alexa voice service* dilakukan beberapa pendefinisian yang nantinya akan digunakan pada saat perintah suara diucapkan. Pada *alexa voice service* didefinisikan *product name*, *product id*, *product category* dan lain sebagainya. Product yang dimaksud tersebut adalah perangkat raspberry pi yang bertindak sebagai modul speech recognition. *Alexa voice service* merupakan perantara antara sub sistem mikrofon dengan alexa skills kit. Gambar 5.11 memperlihatkan alur kerja dari alexa voice service.



**Gambar 5. 11 Alur Kerja Pada Alexa Voice Service**

#### 5.1.9 Perancangan Alexa Skills Kit

Dalam *alexa voice service* perlu dilakukan beberapa hal diantaranya mendefinisikan *invocation name*, *interaction model* dan *service end point type*. *Invocation name* merupakan pendefinisian nama *skill* yang nantinya akan disebutkan saat awal penggunaan sistem. Dalam sistem ini, *invocation name* diberi nama **home security model**. Ketika *invocation name* disebut, maka sistem akan diarahkan ke *skill* tersebut, sehingga sistem dapat menerjemahkan beberapa perintah yaitu untuk membaca data yang didapatkan dari sensor.

Selanjutnya adalah *interaction model*. Perancangan *interaction model* adalah merancang *intent schema* (representasi dari sebuah tindakan yang dapat memenuhi permintaan pengucap) dan *sample utterances* (contoh pengucapan). Selanjutnya adalah *interaction model*, perancangan pada *interaction model* adalah mendefinisikan *intent schema* (representasi tindakan yang dapat memenuhi permintaan pengucap), *custom slot* (argument atau kata kunci dari kalimat pengucap) dan *sample utterances* (contoh pengucapan). *Slot* yang didefinisikan dalam *Alexa Skills Kit* adalah nama dari masing-masing sensor seperti **Ultrasonic** yang berfungsi untuk memberitahukan hasil data yang didapatkan dari sensor ultrasonik, **Microwave** yang berfungsi untuk memberitahukan hasil data yang didapatkan dari sensor microwave, **PIR** yang berfungsi untuk memberitahukan hasil data yang didapatkan dari sensor PIR, dan **Vibration** untuk memberitahukan hasil data yang didapatkan dari sensor Vibration. *Intent schema* di desain dengan bentuk JSON seperti terlihat pada Gambar 5.12.



```

{
  "slots": [
    {
      "name": "nama_slot_1",
      "type": "TIPE_SLOT"
    },
    {
      "name": "nama_slot_2",
      "type": "TIPE_SLOT"
    }
  ],
  "intent": "NamaIntent"
},

```

**Gambar 5. 12 Perancangan Intent Schema**

Pada perancangan *custom slot*, didefinisikan *slot type* dan *value* yang merepresentasikan dari slot tersebut. *Slot type* merupakan sebuah nama yang mewakili *value* yang terdapat di dalamnya. Sedangkan *value* merupakan nilai yang merepresentasikan *slot type* tersebut. Model pendefinisian dapat dilihat pada Gambar 5.13.

Enter Type  
TIPE\_SLOT

Enter Values  
Values must be line-separated

- 1 value1
- 2 value2
- 3 value3
- 4 value4

**Gambar 5. 13 Perancangan Custom Slot**

Sedangkan untuk *sample utterance*, didefinisikan beberapa contoh pengucapan yang mendukung tujuan sistem sehingga sensor dapat mendeteksi sebuah aktivitas atau pergerakan. Dalam sistem ini, didefinisikan beberapa contoh pengucapan, yaitu pengucapan nama *invocation name* **Home security**, kemudian dilanjutkan dengan mengucapkan *slot* yang telah didefinisikan, yaitu nama sensor seperti **Microwave**. Cara pendefinisian *sample utterance* dapat dilihat pada Gambar 5.14.

```

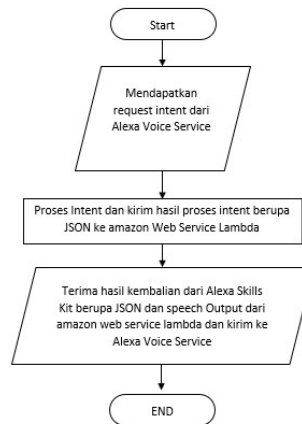
NamaIntent kalimat bebas {nama_slot_1} kalimat bebas {nama_slot_2}
NamaIntent {nama_slot_1} kalimat bebas {nama_slot_2}
NamaIntent kalimat bebas {nama_slot_1} {nama_slot_2}

```

**Gambar 5. 14 Perancangan Sample Utterances**

Selanjutnya adalah *service end point type*. *Service end point type* pada sistem ini merupakan alamat dari *Amazon Web Service Lambda*. Nantinya hasil terjemahan suara pada *Alexa Skills Kit* yang berbentuk JSON akan dikirimkan ke alamat *Amazon Web Service Lambda* dan diolah sehingga data yang di *subscribe*

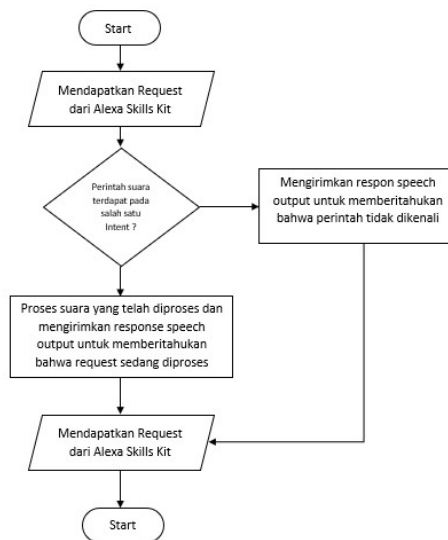
dari MQTT sudah menjadi bentuk lain yang dikenali oleh sistem. Alur yang terdapat pada *Alexa Skills Kit* ini dapat dilihat di *flowchat* pada Gambar 5.15.



**Gambar 5. 15 Alur Kerja Pada Alexa Skills Kit**

#### 5.1.10 Perancangan Amazon Web Service

Pada tahap perancangan *Amazon Web Service Lambda* langkah yang dilakukan adalah membuat sebuah kode program yang nantinya akan menangani terjemahan *intent* dari *Alexa Skills Kit*. Kode program akan dibuat dengan Bahasa pemrograman Node.js. Kemudian program tersebut akan di upload ke *Amazon Web Service Lambda* dan di-*compile*. Program tersebut memiliki kemampuan menerjemahkan *intent request*, mengambil nilai slot, memodelkan data yang di dapat menjadi bentuk yang dikenali oleh sistem. Alur kerja pada *Amazon Web Service Lambda* dapat dilihat pada Gambar 5.16.



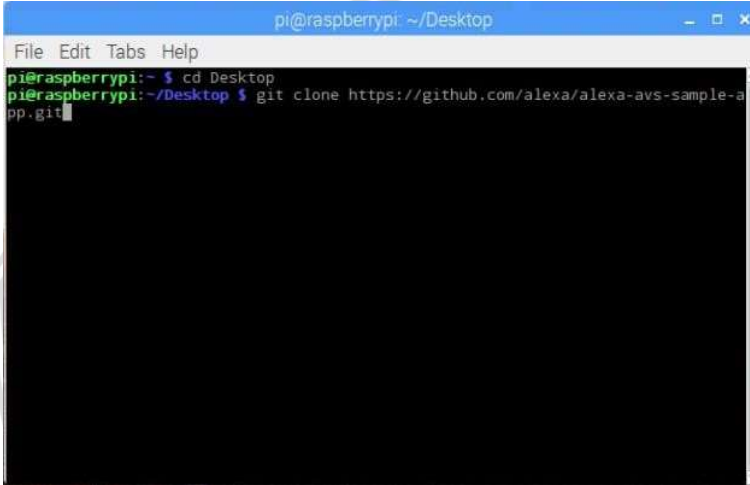
**Gambar 5. 16 Alur Kerja Pada Amazon Web Service Lambda**

## 5.2 Implementasi

### 5.2.1 Implementasi Sub Sistem Mikrofon (Modul Speech Recognition)

Pada bagian implementasi sub sistem mikrofon dilakukan dua hal yaitu pemasangan mikrofon dan pemasangan service alexa paada perangkat raspberry pi. Pemasangan mikrofon cukup dengan menghubungkan mini usb mikrofon ke port usb yang terdapat pada perangkat raspberry pi. Untuk pemasang service alexa dapat dilihat melalui tahap-tahap berikut.

Tahap awal pemasangan service alexa pada perangkat raspberry pi adalah dengan membuka terminal pada perangkat raspberry dan mengetikan perintah seperti Gambar 5.17.



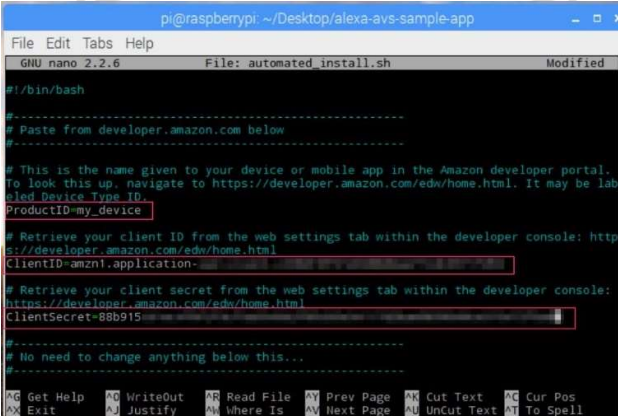
```

pi@raspberrypi: ~/Desktop
File Edit Tabs Help
pi@raspberrypi:~$ cd Desktop
pi@raspberrypi:~/Desktop$ git clone https://github.com/alexa/alexa-avs-sample-app.git

```

Gambar 5. 17 Cloning Alexa AVS dari Github

Setelah proses cloning selesai lakukan konfigurasi pada script instalasi. Bagian yang dikonfigurasi antara lain **Product ID**, **Client ID** dan **ClientSecret**. Tetap pada terminal yang sama, jalankan perintah **cd ~/Desktop/alexa-avs-sample-app** lalu **nano automated\_install.sh**. Akan muncul tampilan seperti Gambar 5.18.



```

pi@raspberrypi: ~/Desktop/alexa-avs-sample-app
File Edit Tabs Help
GNU nano 2.2.6 File: automated_install.sh Modified
#!/bin/bash
# Paste from developer.amazon.com below
#-----
# This is the name given to your device or mobile app in the Amazon developer portal.
# To look this up, navigate to https://developer.amazon.com/edw/home.html. It may be labeled Device Type ID.
ProductID=my_device
# Retrieve your client ID from the web settings tab within the developer console: https://developer.amazon.com/edw/home.html
ClientID=amzn1.application-...
# Retrieve your client secret from the web settings tab within the developer console: https://developer.amazon.com/edw/home.html
ClientSecret=88b915...
# No need to change anything below this...
#-----
AC Get Help AC WriteOut AR Read File AY Prev Page AN Cut Text AC Cur Pos
AX Exit AU Justify AW Where Is AV Next Page AU UnCut Text AT To Spell

```

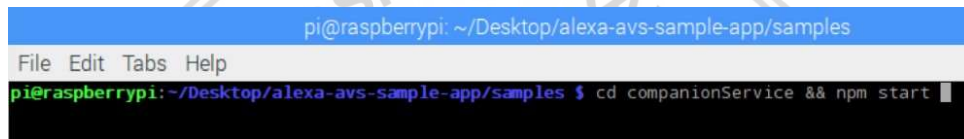
Gambar 5. 18 Konfigurasi Script Instalasi

Selanjutnya setelah script telah selesai di konfigurasi jalankan script dengan perintah **. automated\_install.sh**. Tunggu hingga proses instalasi selesai. Setelah proses instalasi selesai hal selanjutnya yang harus dilakukan adalah menjalankan web service, sample app dan wake word engine. Fungsi dari web service adalah untuk melakukan otorisasi dengan cloud alexa, sample app untuk berkomunikasi ke alexa voice service dan wake word engine supaya sistem dapat memulai interaksi dengan menggunakan nama sebutan tertentu untuk sistem ini adalah alexa.

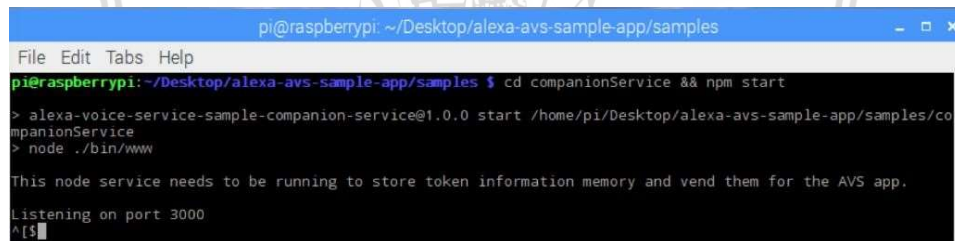
Untuk menjalankan web service buka satu terminal baru lalu jalankan perintah berikut :

```
cd ~/Desktop/alexa-avs-sample-app/samples cd companionService && npm start
```

Maka tampilan terminal tersebut akan terlihat seperti Gambar 5.19 dan Gambar 5.20.



Gambar 5. 19 Web Service Bagian 1

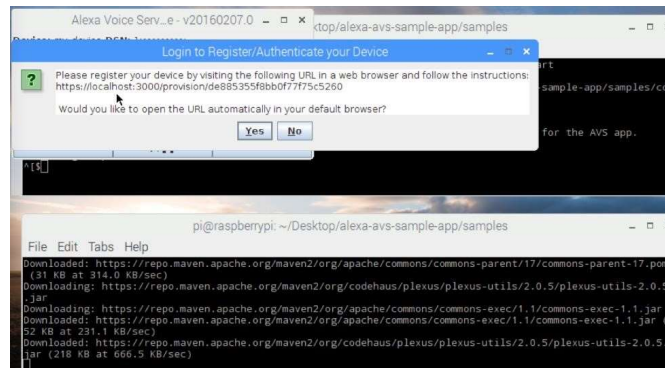


Gambar 5. 20 Web Service Bagian 2

Selanjutnya buka terminal kedua, pada terminal ini akan dijalankan sample app dengan perintah sebagai berikut :

```
Cd~/Desktop/alexa-avs-sample-app/samples cd javaclient && mvn exec:exec
```

Maka tampilan terminal tersebut akan terlihat seperti Gambar 5.21. Selanjutnya ketika ditekan tombol yes maka halaman sebuah halaman web browser untuk login ke akun amazon akan terbuka. Lakukan login supaya sistem dapat terhubung ke alexa voice service seperti pada Gambar 5.22.



Gambar 5. 21 Sample App Bagian 1



Sign in to My Alexa Voice Service Sample App Security using your Amazon account

E-mail or mobile number:

What is your password?

☐ Keep me signed in. [Details](#)

[Forgot your password?](#)  
[Create an Amazon.com account.](#)

**Login without hassle**  
 Use Amazon to log into this site without another password. ☒

**Login safely**  
 Amazon does not share your password with this site.

[Learn More](#)

Gambar 5. 22 Login Akun Amazon

Setelah login akan muncul halaman konfirmasi autentikasi pilih Okay seperti Gambar 5.23, lalu tampilan browser akan terlihat seperti Gambar 5.24. Selanjutnya tekan OK seperti Gambar 5.25, maka sistem akan siap menerima input suara yang nantinya akan diteruskan ke Alexa Voice Service.



Hi Not ?

When you click "Okay", we'll provide **Alexa Voice Service Sample App Security Profile**:

- Connectivity to Alexa so that you can interact with it using your voice.

By clicking "Okay", you also accept: All Amazon terms found [here](#)

Alexa processes and retains audio and other information in the cloud to provide and improve our

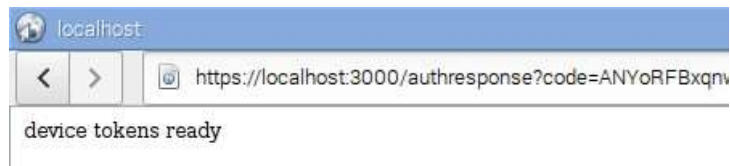
To remove access, visit [Your Account](#) at Amazon.

**Login without hassle**  
 Use Amazon to log into this site without another password. ☒

**Login safely**  
 Amazon does not share your password with this site.

[Learn More](#)

Gambar 5. 23 Konfirmasi Autentikasi

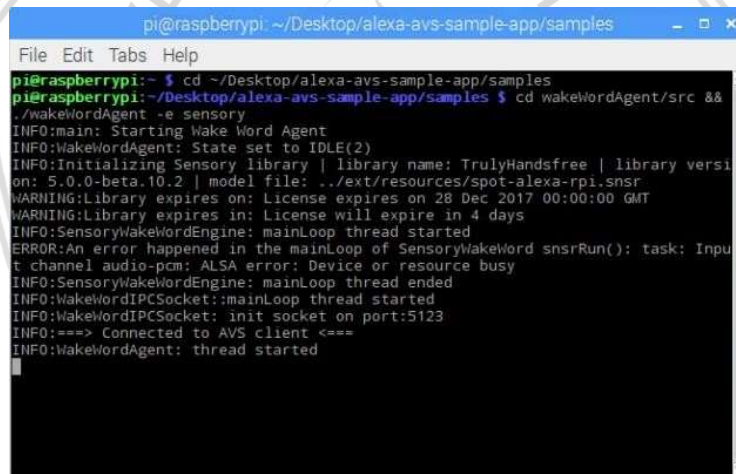


Gambar 5. 24 Tampilan Browser Setelah Konfirmasi Autentikasi



Gambar 5. 25 Konfirmasi Registrasi Sukses

Selanjutnya adalah bagian wake word engine, seperti tahap sebelumnya, buka satu terminal baru lalu jalankan perintah berikut seperti Gambar 5.26 `cd ~/Desktop/alexa-avs-sample-app/samples` `cd wakeWordAgent/src` && `./wakeWordAgent -e sensory`.



Gambar 5. 26 Wake Word Engine

Setelah ketiga hal tersebut dijalankan maka sistem akan siap menerima perintah suara yang akan diteruskan ke alexa voice service.

### 5.2.2 Implementasi Sub Sistem Sensor PIR

Pada implementasi sub sistem *PIR* dilakukan dua hal yaitu melakukan pengkoneksian modul komunikasi nRF24L01 dan sensor *PIR* ke *Arduino Uno Board*. Pada sub sistem ini, modul komunikasi nRF24L01 berfungsi sebagai media komunikasi antara *Arduino Uno Board* dengan Raspberry Pi dan juga sebagai media pengiriman data. Sedangkan sensor *PIR* berfungsi untuk mendeteksi sebuah gerakan. Data yang didapatkan dari sensor *PIR* merupakan data Digital dengan *value* 0 dan 1. Jika data yang diterima bernilai 0, maka tidak ada gerakan yang terdeteksi oleh sensor. Jika data yang diterima bernilai 1, maka sensor



mendeteksi sebuah gerakan. Kode program dibuat dengan menggunakan bahasa pemrograman Arduino, sehingga sensor dan modul komunikasi nRF24L01 dapat bekerja sesuai dengan kode program. Pada tahap implementasi ini akan diperlihatkan potongan-potongan kode program berdasarkan fungsinya pada tabel-tabel berikut :

**Tabel 5. 6 Pendefinisian Library dan Variable Sub Sistem Sensor *PIR***

No	Source Code
1	#include<SPI.h>
2	#include<RF24.h>
3	#include<nRF24L01.h>
4	
5	RF24 radio(9, 10);
6	int calibrationTime = 15;
7	long unsigned int lowIn;
8	long unsigned int pause = 5000;
9	boolean lockLow = true;
10	boolean takeLowTime;
11	int led = 13;
12	int inputpin = 6;
13	int val=0;
14	const uint64_t wadd = 0xF0F0F0F0D1LL;
15	const uint64_t radd = 0xF0F0F0F0B1LL;

Pada Table 5.6 merupakan potongan program pendefinisian library dan variable pendukung sub sistem. Library yang digunakan adalah SPI, RF24 dan nRF24L01. Library SPI berfungsi untuk menangani komunikasi serial di arduino. Library RF24 dan nRF24L01 berfungsi untuk komunikasi jarak jauh atau nirkabel dengan memanfaatkan gelombang RF 2.4 GHz.

**Tabel 5. 7 Pendefinisian Kondisi Awal Sub Sistem Sensor *PIR***

No	Source Code
1	radio.begin();
2	radio.setPALevel(RF24_PA_MAX);
3	radio.setChannel(0x76);
4	radio.openWritingPipe(wadd);
5	radio.openReadingPipe(1, radd);
6	radio.enableDynamicPayloads();
7	radio.stopListening();
8	radio.powerUp();
9	Serial.begin(9600);
10	pinMode(inputpin, INPUT);
11	pinMode(led, OUTPUT);
12	digitalWrite(inputpin, LOW);

Pada Table 5.7 merupakan potongan kode program dimana kondisi awal sistem di definisikan, yaitu sensor akan berada pada kondisi *standby* dan bersiap untuk mendeteksi gerak. Dalam potongan kode program tersebut, terdapat beberapa kode yang berfungsi untuk mendefinisikan *pipe address* untuk menghubungkan modul nRF24L01 yang terhubung di *Arduino Uno Board* dengan modul nRF24L01 yang terhubung dengan Raspberry Pi sebagai broker MQTT.

**Tabel 5. 8 Pendefinisian Pengiriman data Sub Sistem Sensor *PIR***

No	Source Code
1	<code>if(digitalRead(inputpin) == HIGH){</code>
2	<code>  if(lockLow){</code>
3	<code>    lockLow = false;</code>
4	<code>    const char text1[] = "1";</code>
5	<code>    radio.write(&amp;text1, sizeof(text1));</code>
6	<code>    //Serial.println("1");</code>
7	<code>    Serial.print("motion detected at ");</code>
8	<code>    Serial.print(millis()/1000);</code>
9	<code>    Serial.println(" sec");</code>
10	<code>    digitalWrite(led,HIGH);</code>
11	<code>    //delay(50);</code>
12	<code>  }</code>
13	<code>  takeLowTime = true;</code>
14	<code>}</code>
15	
16	<code>if(digitalRead(inputpin) == LOW){</code>
17	<code>  if(takeLowTime){</code>
18	<code>    lowIn = millis();</code>
19	<code>    takeLowTime = false;</code>
20	<code>  }</code>
21	<code>  if(!lockLow &amp;&amp; millis() - lowIn &gt; pause){</code>
22	<code>    lockLow = true;</code>
23	<code>    const char text2[] = "0";</code>
24	<code>    radio.write(&amp;text2, sizeof(text2));</code>
25	<code>    Serial.println("0");</code>
26	<code>    Serial.print("motion ended at ");</code>
27	<code>    Serial.print((millis() - pause)/1000);</code>
28	<code>    Serial.println(" sec");</code>
29	<code>    digitalWrite(led,LOW);</code>
30	<code>    //delay(50);</code>
31	<code>  }</code>
32	<code>}</code>

Pada Table 5.8 merupakan potongan program yang mendefinisikan pendeteksian pergerakan dan pengiriman data dari Arduino ke Raspberry Pi menggunakan modul komunikasi nRF24L01. Kode program tersebut memanfaatkan fungsi char array untuk mengirimkan data yang didapatkan dari sensor *PIR*. Kemudian data akan diolah per-karakter didalam Raspberry Pi kemudian data akan dikonversi kedalam bentuk String yang nantinya akan di *publish* ke MQTT.

### 5.2.3 Implementasi Sub Sistem Sensor Ultrasonik

Pada implementasi sub sistem Sensor Ultrasonik, dilakukan 2 hal yaitu melakukan pengkoneksian antara modul nRF24L01 yang terhubung di Arduino sebagai transceiver dengan modul nRF24L01 yang terhubung pada Raspberry Pi sebagai receiver dan pengkoneksian sensor ultrasonik dengan *Arduino Uno Board* sebagai sensor untuk mendeteksi jarak. Data yang didapatkan dari sensor ini merupakan data float yang nantinya akan diubah menjadi sebuah char array agar

dapat dikirimkan ke broker MQTT melalui modul nRF24L01 dan kemudian di *publish* ke MQTT. Pada tahap implementasi ini, akan diperlihatkan potongan-potongan kode program pada table berdasarkan fungsinya.

**Tabel 5. 9 Pendefinisian Library dan Variable Sub Sistem Sensor Ultrasonik**

No	Source Code
1	#include<SPI.h>
2	#include<RF24.h>
3	
4	RF24 radio(9, 10);
5	#define trigPin 3
6	#define echoPin 4
7	const uint64_t wadd = 0xF0F0F0F0D4LL;
8	const uint64_t radd = 0xF0F0F0F0B4LL;
9	double jarak;
10	String str;
11	char cstr[16];

Pada table 5.9 merupakan potongan-potongan kode program untuk mendefinisikan library yang dibutuhkan oleh sistem dan mendefinisikan beberapa variable pendukung sistem. Dalam potongan kode program ini, terdapat library SPI yang berfungsi untuk menangani komunikasi serial pada Arduino uno. Sedangkan library RF24, berfungsi untuk komunikasi jarak jauh atau nirkabel dengan memanfaatkan gelombang RF 2.4 GHz.

**Tabel 5. 10 Pendefinisian Kondisi Awal Sub Sistem Sensor Ultrasonik**

No	Source Code
1	radio.begin();
2	radio.setPALevel(RF24_PA_MAX);
3	radio.setChannel(0x76);
4	radio.openWritingPipe(wadd);
5	radio.openReadingPipe(1, radd);
6	radio.enableDynamicPayloads();
7	radio.powerUp();
8	pinMode(trigPin, OUTPUT);
9	pinMode(echoPin, INPUT);
10	Serial.begin(9600);

Pada table 5.10 merupakan potongan kode program untuk mengatur kondisi awal sebelum sistem dapat berjalan sesuai keinginan. Dalam potongan kode tersebut, terdapat kode program untuk mengatur alamat pipe pada modul nRF24L01 yang berfungsi sebagai media komunikasi antara Arduino dengan Raspberry Pi. Dalam potongan kode program tersebut juga terdapat beberapa kode program untuk mengatur sebuah Input dan Output dari sistem ini.

**Tabel 5. 11 Pendefinisian Pengiriman data Sub Sistem Sensor *PIR***

No	Source Code
1	double duration, distance;
2	digitalWrite(trigPin, LOW);
3	delayMicroseconds(2);
4	digitalWrite(trigPin, HIGH);
5	delayMicroseconds(10);
6	duration = pulseIn(echoPin, HIGH);
7	distance = (duration/2) / 29.1;
8	if (distance >= 8){
9	Serial.print("Jendela Terbuka dengan jarak Kerapatan : ");
10	Serial.print(distance);
11	Serial.println(" Cm");
12	str = String(distance);
13	str.toCharArray (cstr, 16);
14	radio.write(&cstr, sizeof(cstr));
15	
16	} else if (distance <= 8){
17	Serial.print("Jendela Masih Tertutup, kerapatan Jendela : ");
18	Serial.print(distance);
19	Serial.println(" Cm");
20	const char text2[] = "000";
21	radio.write(&text2, sizeof(text2));
22	delay(1000);
23	
24	}

Pada table 5.11 merupakan potongan kode program yang berfungsi untuk mendefinisikan pengiriman data dari sub sistem sensor *Ultrasonic* ke Raspberry Pi. Data yang diterima dari sensor ultrasonik bertipe data Integer, kemudian sebelum dikirim ke Raspberry Pi, data akan diubah menjadi sebuah tipe data char array agar data yang dikirimkan tetap dan tidak berubah. Setelah data sampai di Raspberry Pi, maka data akan langsung dikumpulkan dan di ubah menjadi String yang nantinya data akan langsung di *Publish* ke MQTT.

#### 5.2.4 Implementasi Sub Sistem Sensor *Vibration*

Implementasi pada sub sistem sensor *Vibration*, dilakukan 2 hal yaitu pengkoneksian antara modul nRF24L01 sebagai transceiver dan sensor *vibration* pada *Arduino Uno Board*. Dalam sub sistem ini, modul nRF24L01 berfungsi sebagai modul komunikasi dan pengiriman data dari Arduino menuju Raspberry Pi. Sedangkan sensor *Vibration* berfungsi untuk mendeteksi sebuah getaran. Sensor ini dapat ditempatkan pada sebuah pintu ataupun jendela, sehingga jika pintu ataupun jendela terbuka, maka getaran tersebut memicu sensor ini untuk dapat bekerja. Data yang didapatkan dari sensor ini berbentuk digital, yaitu 0 dan 1. Jika *value* bernilai 0, maka sensor tidak mendeteksi getaran. Namun jika sensor bernilai

1, maka sensor mendeteksi adanya getaran. Kemudian data yang diperoleh oleh sensor akan diubah menjadi sebuah tipe data char-array dan langsung dikirimkan ke broker MQTT yang nantinya akan di publish ke MQTT melalui Raspberry Pi.

**Tabel 5. 12 Pendefinisian Library dan Variable Sub Sistem Sensor *Vibration***

No	Source Code
1	#include<SPI.h>
2	#include<RF24.h>
3	#include<nRF24L01.h>
4	
5	RF24 radio(9, 10);
6	int vib_pin=8;
7	int calibrationTime = 15;
8	long unsigned int lowIn;
9	long unsigned int pause = 5000;
10	boolean lockLow = true;
11	boolean takeLowTime;
12	int led_pin = 13;
13	const uint64_t wadd = 0xF0F0F0F0D2LL;
14	const uint64_t radd = 0xF0F0F0F0B2LL;

Pada table 5.12 merupakan potongan-potongan kode program yang terdiri dari beberapa Library dan Variabel pendukung sub sistem ini. Dalam kode program tersebut, terdapat beberapa library yaitu SPI.h, RF24.h dan nRF24L01.h. Library SPI.h berfungsi sebagai media komunikasi serial pada board Arduino. Sedangkan Library RF24.h dan nRF24L01.h berfungsi sebagai media komunikasi nirkabel yang memanfaatkan gelombang radio 2.4 GHz. Di potongan kode program tersebut juga mendefinisikan sebuah alamat pipe untuk berkomunikasi antara modul nRF24L01 yang berfungsi sebagai Transciever dengan modul nRF24L01 yang berfungsi sebagai Receiver.

**Tabel 5. 13 Pendefinisian Kondisi awal sub sistem sensor *Vibration***

No	Source Code
1	radio.begin();
2	radio.setPALevel(RF24_PA_MAX);
3	radio.setChannel(0x76);
4	radio.openWritingPipe(wadd);
5	radio.openReadingPipe(1, radd);
6	radio.enableDynamicPayloads();
7	radio.powerUp();
8	Serial.begin(9600);
9	pinMode(vib_pin, INPUT);
10	pinMode(led_pin, OUTPUT);

Pada table 5.13 merupakan potongan kode program untuk mendefinisikan beberapa pengaturan untuk sub sistem ini. Dalam potongan kode program tersebut, terdapat beberapa pengaturan seperti mendefinisikan alamat pipe untuk menghubungkan modul nRF24L01 yang berfungsi sebagai transceiver dengan modul nRF24L01 sebagai receiver.

**Tabel 5. 14 Pendefinisian Pengiriman Data Pada Sub Sistem Sensor *Vibration***

No	Source Code
1	int val;
2	val=digitalRead(vib_pin);
3	if(val==1)
4	{
5	digitalWrite(led_pin,HIGH);
6	Serial.println("Ada Orang Masuk");
7	const char text1[] = "01";
8	radio.write(&text1, sizeof(text1));
9	delay(1000);
10	}
11	else if (val == 0){
12	digitalWrite(led_pin,LOW);
13	Serial.println("Rumah Aman");
14	//const char text2[] = "00";
15	//radio.write(&text2, sizeof(text2));
16	delay(1000);
17	}

Pada table 5.14 merupakan potongan kode program yang berfungsi untuk mendefinisikan pengiriman data dari Sub Sistem Sensor *Vibration* ke MQTT Broker dan untuk mendeteksi getaran melalui sensor *Vibration*. Dalam potongan kode program ini, data yang didapatkan oleh Sensor *Vibration* berupa data digital, yaitu 0 dan 1. Data yang telah didapat kemudian akan diubah menjadi sebuah tipe data char-array yang kemudian akan dikirimkan ke MQTT Broker (Raspberry Pi) dengan menggunakan modul nRF24L01.

### 5.2.5 Implementasi Sub Sistem Sensor *Microwave*

Implementasi pada Sub Sistem Sensor *Microwave* dilakukan 3 hal, yaitu pengkoneksian antara sensor *microwave* dengan rangkaian penguat sinyal, pengkoneksian antara sensor *Microwave* dan penguat sinyal ke *Arduino uno board*, dan pengkoneksian antara modul komunikasi nRF24L01 dengan *Arduino uno board*. Pada sub sistem ini, sensor *microwave* terhubung di sebuah rangkaian penguat sinyal yang dilengkapi dengan menggunakan sebuah modul LM324 sebagai perantara antara input dan output. Data yang didapatkan oleh sensor *microwave* bertipe data float yang nantinya akan di ubah menjadi char-array agar data dapat dikenali per-karakter dan kemudian dapat langsung dikirm menuju broker MQTT. Modul nRF24L01 berfungsi sebagai media komunikasi antara node



(transciever) dengan Raspberry Pi (receiver) dan juga berfungsi untuk mengirimkan data yang telah didapatkan dari sensor *microwave*. Data yang telah dikirimkan ke MQTT Broker (Raspberry Pi) kemudian akan langsung dipublish ke MQTT .

**Tabel 5. 15 Pendefinisian Library dan Variable Sub Sistem Sensor *Microwave***

No	Source Code
1	#include "FreqPeriod.h"
2	#include<SPI.h>
3	#include<RF24.h>
4	#include<nRF24L01.h>
5	
6	double lfrq;
7	double spd;
8	long int pp;
9	RF24 radio(9, 10);
10	const uint64_t wadd = 0xF0F0F0F0D3LL;
11	const uint64_t radd = 0xF0F0F0F0B3LL;
12	String str;
13	char cstr[16];
14	double kecepatan;

Pada Tabel 5.15 merupakan potongan kode program yang berfungsi untuk mendefinisikan library dan variable pendukung pada sub sistem ini. Dalam potongan kode program ini, terdapat beberapa library yaitu FreqPeriod, SPI, RF24 dan nRF24L01. Library FreqPeriod.h berfungsi untuk mengukur frekuensi yang diterima oleh gelombang radio yang dipancarkan oleh sensor *Microwave*. Library SPI berfungsi sebagai media komunikasi secara serial di *Arduino Uno board*. Sedangkan Library RF24 dan nRF24L01 berfungsi sebagai media komunikasi dan pengiriman data antara transceiver dengan receiver.

**Tabel 5. 16 Pendefinisian Kondisi awal sub sistem sensor *Microwave***

No	Source Code
1	radio.begin();
2	radio.setPALevel(RF24_PA_MAX);
3	radio.setChannel(0x76);
4	radio.openWritingPipe(wadd);
5	radio.openReadingPipe(1, radd);
6	radio.enableDynamicPayloads();
7	radio.powerUp();
8	Serial.begin(9600);
9	FreqPeriod::begin();



Pada table 5.16 merupakan potongan program untuk mendefinisikan pengaturan dari sub sistem ini. Dalam potongan program ini, terdapat beberapa pengaturan yaitu pengaturan *pipe address* untuk menghubungkan modul nRF24L0 sebagai transceiver dengan modul nRF24L01 sebagai Receiver dan pengaturan sensor *microwave*.

**Tabel 5. 17 Pendefinisian Pengiriman Data Pada Sub Sistem Sensor *Vibration***

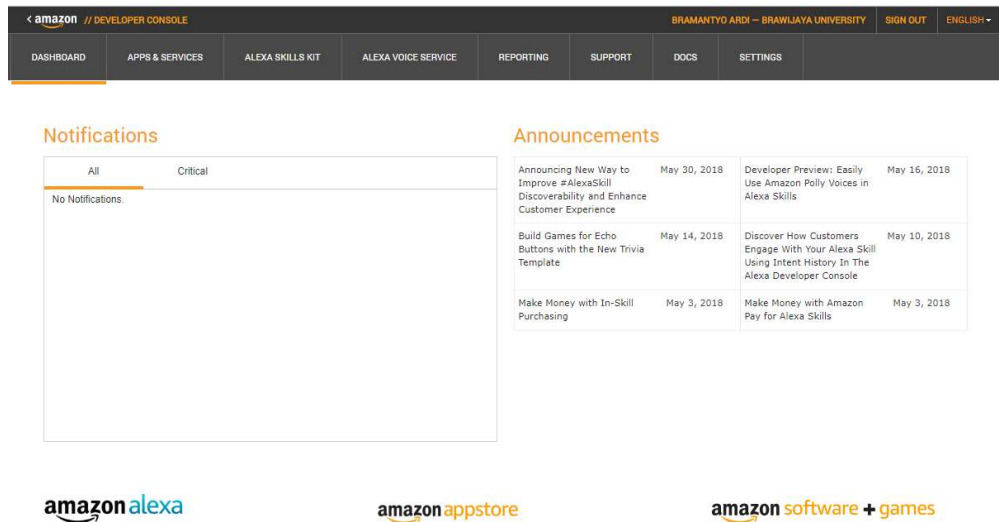
No	Source Code
1	pp = FreqPeriod::getPeriod();
2	if (pp) {
3	lfrq = 16000400.0 /pp;
4	spd = lfrq/31.36;
5	if (lfrq < 0){
6	Serial.println ("Tidak ada Gerakan");
7	} else if (lfrq >= 0){
8	kecepatan = lfrq/31.36;
9	Serial.print("Ada Gerakan dengan Kecepatan : ");
10	Serial.print(kecepatan);
11	Serial.println(" Mph");
12	str = String(kecepatan);
13	str.toCharArray (cstr, 16);
14	radio.write(&cstr, sizeof(cstr));
15	}
16	}

Pada table 5.17 merupakan potongan kode program yang berfungsi untuk mendefinisikan pengiriman data dari sub sistem sensor *Microwave* ke Raspberry Pi. Data yang diterima dari sensor *Microwave* bertipe float, kemudian sebelum dikirim ke Raspberry Pi, data akan diubah menjadi sebuah tipe data char array agar data yang dikirimkan tetap dan tidak berubah. Setelah data sampai di Raspberry Pi, maka data akan langsung dikumpulkan dan di ubah menjadi String yang nantinya data akan langsung di *Publish* ke MQTT.

### 5.2.6 Implementasi Alexa Voice Service

Pada tahap implementasi *alexa voice service* akan dijelaskan tahap-tahap pengaturan yang dilakukan supaya *alexa voice service* dapat bekerja sesuai dengan yang diharapkan. Implementasi *alexa voice service* dilakukan dengan tahapan-tahapan sebagai berikut :

Hal pertama yang dilakukan adalah melakukan login pada halaman *developer amazon* yang beralamat di : <http://developer.amazon.com> lalu masuk ke halaman *dashboard alexa*. Klik *Get Started* seperti Gambar 5.27 berikut.



**Gambar 5. 27 Tampilan Dashboard Alexa**

Selanjutnya adalah tekan tombol create product maka halaman website akan diarahkan ke halaman product information dimana informasi dari product atau device yang akan digunakan harus didefinisikan disini seperti product name, product id, product category, product description dan beberapa hal lain yang dapat dilihat pada Gambar 5.28, Gambar 5.29 dan Gambar 5.30 berikut.

**Gambar 5. 28 Product Information Alexa Voice Service 1**

Welcome to Alexa Voice

Secure | <https://developer.amazon.com/avs/home.html#/avs/products/new>

Brief product description \*

Produk ini digunakan untuk mengontrol lampu menggunakan perintah suara

How will end users interact with your product? \*

☐ Touch-initiated  
A user's primary way to interact with Alexa is by tapping or holding a button.

☒ Hands-free  
Hands-free products allow users to interact with Alexa by using their voice at a close distance.

☐ Far-field  
Far-field products allow users to interact with Alexa by using their voice from a long distance.

Upload an image  
This image is shown to end users on the [Manage your Content and Devices](#) page.

UPLOAD IMAGE  
MAX: 142 X 130 PX

**Gambar 5. 29 Product Information Alexa Voice Service 2**

Welcome to Alexa Voice

Secure | <https://developer.amazon.com/avs/home.html#/avs/products/new>

Do you intend to distribute this product commercially? \*

☐ Yes

☒ No

Is this a children's product or is it otherwise directed to children younger than 13 years old? \* [Learn More](#)

☐ Yes

☒ No

NEXT

FAQs | Contact Us | App Distribution Agreement | Trademark Guidelines | Terms of Use | Job Opportunities

**Gambar 5. 30 Product Information Alexa Voice Service 3**

Selanjutnya setelah selesai pada pengisian product information pada alexa voice service klik next maka halaman website akan diarahkan ke halaman LWA Security Profile. Pada halaman tersebut dilakukan pendefinisian security profile dan platform information seperti Gambar 5.31 dan Gambar 5.32 berikut.

Select a Security Profile

A security profile associates user data and security credentials with one or more related products.

Security Profile \*

Alexa Security

or [CREATE NEW PROFILE](#)

Security Profile description

Alexa Security

Security Profile ID ⓘ

amzn1.application.431a522993ff419a8bfae83f63877508 [COPY](#)

Gambar 5. 31 Security Profile Alexa Voice Service

Platform information

Specify the settings for the websites or mobile apps that will use Login with Amazon with the selected Security Profile.

Web Android / Kindle iOS Other devices and platforms

Add all possible origin URLs of your LWA implementation to associate with the web Client ID and secret below. [Learn More](#)

Client ID ⓘ

amzn1.application-aa2-client.077066947c1046d19570daf7cc1dd53b [COPY](#)

Client secret ⓘ

1ec6806b5b9397c58beea85a3a7f13089da006fcaadbdb4c1397960a3ec5b90 [COPY](#)

Allowed origins ⓘ

<https://www.example.com> [ADD](#)

<https://localhost:3000> [X](#)

Allowed return URLs ⓘ

<https://www.example.com/login.php> [ADD](#)

<https://localhost:3000/authresponse> [X](#)

Gambar 5. 32 Platform Information Alexa Voice Service

Hal penting dari pendefinisian di atas adalah pada bagian **Security Profile ID**, **Client ID** dan **Client Secret** dimana bagian tersebut nantinya akan digunakan untuk implementasi sub sistem mikrofon sebagai modul *speech recognition* supaya sistem dapat terhubung ke *alexa voice service*.

### 5.2.7 Implementasi Alexa Skills Kit

Pada tahap implementasi alexa skills kit akan dijelaskan tahap-tahap pengaturan yang dilakukan supaya alexa skills kit dapat bekerja sesuai dengan yang diharapkan. Implementasi alexa skills kit dilakukan dengan tahapan-tahapan sebagai berikut :

Seperti pada alexa voice service, lakukan login pada halaman developer amazon, lalu arahkan halaman website ke dashboard alexa lalu klik getting started pada alexa skills kit seperti terlihat pada Gambar 5.27. Lalu halaman website akan diarahkan halaman alexa skills kit. Pilih add a new skill selanjutnya isikan informasi seperti pada Gambar 5.33 berikut.

**Skill Type**  
Define a custom interaction model or use one of the predefined skill APIs. [Learn more](#)

☒ Custom Interaction Model  
☐ Smart Home Skill API  
☐ Flash Briefing Skill API  
☐ Video Skill API

---

**Language**  
Language of your skill

English (U.S.) ▼

---

**Name**  
Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.

Alexa Smart Lamp

---

**Invocation Name**  
The name customers use to activate the skill. For example, "Alexa ask Tide Pooler..."

lamp control mode

**Gambar 5. 33 Skill Information Alexa Skills Kit**

Selanjutnya adalah pendefinisian pada Intent Schema yang telah dibahas pada bagian perancangan. Intent schema merupakan format standar dari alexa skills kit dalam bentuk JSON dimana format tersebut berfungsi untuk menerjemahkan request yang dikirim dari alexa voice service. Table 5.18 menunjukkan intent schema yang telah di desain untuk sistem ini dan Gambar 5.34 merupakan tampilan kolom pada halaman interaction model untuk bagian intent schema.

**Tabel 5. 18 Intent Schema**

No	Source Code
1	{
2	"interactionModel": {
3	"languageModel": {
4	"invocationName": "home security",
5	"intents": [
6	{
7	"name": "AMAZON.CancelIntent",
8	"samples": []
9	},
10	{
11	"name": "AMAZON.HelpIntent",
12	"samples": []
13	},
14	{
15	"name": "AMAZON.StopIntent",
16	"samples": [
17	"ok thank you"
18	]
19	},
20	{
21	"name": "ReadIntent",
22	"slots": [
23	{
24	"name": "sensor",
25	"type": "Sensor"

Tabel 5. 18 Intent Schema

26	}
27	],
28	"samples": [
29	"read {sensor} sensor"
30	]
31	}
32	],
33	"types": [
34	{
35	"name": "Sensor",
36	"values": [
37	{
38	"name": {
39	"value": "microwave"
40	}
41	},
42	{
43	"name": {
44	"value": "ultrasonic"
45	}
46	},
47	{
48	"name": {
49	"value": "PIR"
50	}
51	},
52	{
53	"name": {
54	"value": "vibration"
55	}
56	}
57	]
58	}
59	] }
60	}
61	}
62	}

#### JSON Editor

[Click here](#) to learn more about the schema definition for interaction models.

Drag and drop a .json file

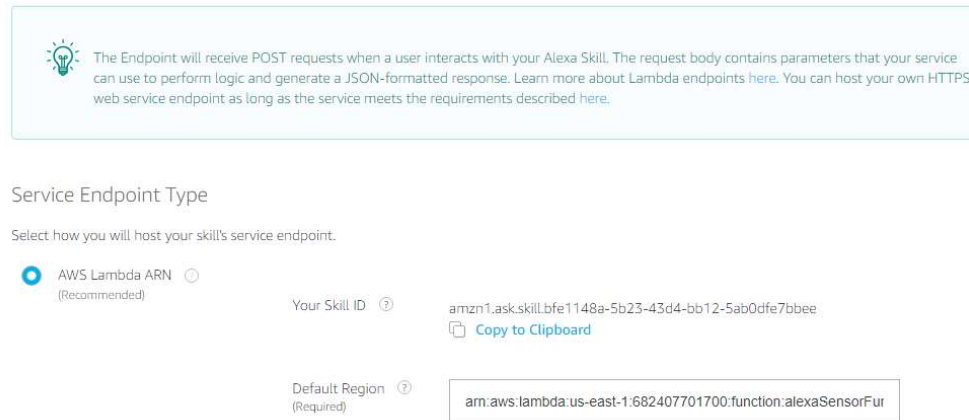
```

1 {
2   "interactionModel": {
3     "languageModel": {
4       "invocationName": "home security",
5       "intents": [
6         {
7           "name": "AMAZON.CancelIntent",
8           "samples": []
9         },
10        {
11          "name": "AMAZON.HelpIntent",
12          "samples": []
13        },
14        {
15          "name": "AMAZON.StopIntent",
16          "samples": [
17            "ok thank you"
18          ]
19        }
20      ]
21    }
22  }
23 }
```

Gambar 5. 34 Kolom Intent Schema

Selanjutnya adalah pada bagian configuration. Pada bagian ini didefinisikan service end point dari alexa skills kit ini. Service end point merupakan tempat dimana output berupa JSON dari alexa skills kit di proses, dalam hal ini output tersebut nantinya akan di proses di Amazon Web Service Lambda maka dari itu alamat dari service end point diarahkan ke alamat Amazon Web Service Lambda yang didapatkan ketika pengimplementasian Amazon Web Service Lambda seperti pada Gambar 5.35 berikut.

### Endpoint






The Endpoint will receive POST requests when a user interacts with your Alexa Skill. The request body contains parameters that your service can use to perform logic and generate a JSON-formatted response. Learn more about Lambda endpoints [here](#). You can host your own HTTPS web service endpoint as long as the service meets the requirements described [here](#).

Service Endpoint Type

Select how you will host your skill's service endpoint.

☒ AWS Lambda ARN (Recommended) ☐

Your Skill ID  amzn1.ask.skill.bfe1148a-5b23-43d4-bb12-5ab0dfe7bbee  [Copy to Clipboard](#)

Default Region  (Required)

Function ARN

**Gambar 5. 35 Pendefinisian Service End Point ke Alamat AWS Lambda**

## 5.2.8 Implementasi Amazon Web Service Lambda

Pada pengimplementasian Amazon Web Service Lambda dilakukan pembuatan kode program berbasis Node.js. Kode program tersebut digunakan untuk menangani request perintah suara. Sehingga ketika user memberikan perintah suara, kemudian AWS Lambda men-subscribe data sensor yang telah di publish ke MQTT melalui Raspberry Pi yang nantinya data sensor tersebut akan dibacakan oleh Alexa . Berikut kode program akan dibagi dalam beberapa bagian yang dapat dilihat pada Gambar berikut.

**Tabel 5. 19 Pendefinisian Library**

No	Source Code
1	const Alexa = require('alexa-sdk');
2	const mqtt = require('mqtt');

**Tabel 5. 20 Pendefinisian Speech Output**

No	Source Code
1	const WELCOME_MESSAGE = 'Welcome to the home security, What do you
2	want me to inform ?';
3	const STOP_MESSAGE = 'Ok no problem, have a good day !';
4	let READ_RESPONSE = '';



**Tabel 5. 21**Pen definisian Opsi Koneksi MQTT Broker

No	Source Code
1	var options = {
2	port: '1883',
3	clientId: 'mqttjs_' + Math.random().toString(16).substr(2, 8),
4	};

**Tabel 5. 22** Launch Request Handler

No	Source Code
1	LaunchRequest': function () {
2	console.log('launch request');
3	this.response.speak(WELCOME_MESSAGE)
4	.listen('Wanna Control Something ?');
5	this.emit(':responseReady');
6	},

**Tabel 5. 23** Subscribe Intent

No	Source Code
1	var mqttPromise = new Promise(function(resolve, reject) {
2	var client = mqtt.connect('mqtt://iot.eclipse.org',
3	options);
4	client.on('connect', function() {
5	client.subscribe('/sensor/' + sensor)
6	client.on('message', (topic, message, packet) =>
7	{
8	var data = packet.retain;
9	client.end();
10	resolve(message.toString());
11	});
12	});
13	});

**Tabel 5. 24** Sensor PIR Intent

No	Source Code
1	if (sensor == 'peer') {
2	if (data == '0'){
3	READ_RESPONSE = 'For now,
4	your peer sensor did not detect any movements';
5	} else {
6	READ_RESPONSE = 'According
7	to the peer sensor, there is movement detected at : ' + data + ' O'clock';
8	}
9	}

**Tabel 5. 25 Sensor *Vibration Intent***

No	Source Code
1	else if (sensor == 'vibration') {
2	READ_RESPONSE = 'According to the
3	Vibration Sensor, it says that there is a malicious activity detected at : ' + data + '
4	O'clock';
5	}

**Tabel 5. 26 Sensor *Ultrasonic Intent***

No	Source Code
1	else if (sensor == 'ultrasonic') {
2	if (data == '000') {
3	READ_RESPONSE = 'According
4	to the ultrasonic sensor, Your Windows is closed ! the distance is below 4
5	centimeters';
6	} else {
7	READ_RESPONSE = 'According
8	to the ultrasonic sensor, Your Windows is open ! the distance is : ' + data + '
9	centimeters ';
10	}
11	}

**Tabel 5. 27 Sensor *Microwave Intent***

No	Source Code
1	else if (sensor == 'microwave') {
2	READ_RESPONSE = 'Your microwave
3	sensor detecting movement with speed : ' + data + ' miles per hour ';
4	}

**Tabel 5. 28 *Help Intent* dan *Stop Intent***

No	Source Code
1	'AMAZON.StopIntent': function () {
2	this.response.speak(STOP_MESSAGE);
3	this.emit('responseReady');
4	},

**Tabel 5. 29 Handler Alexa**

No	Source Code
1	exports.handler = function (event, context, callback) {
2	const alexa = Alexa.handler(event, context, callback);
3	alexa.APP_ID = APP_ID;
4	alexa.registerHandlers(handlers);
5	alexa.execute();
6	};

Pada potongan beberapa kode program tersebut, terdapat kode program yang berfungsi untuk mendefinisikan sebuah perintah yang akan dijalankan oleh Alexa. Dalam kode program ini juga terdapat beberapa pendefinisian untuk men-subscribe data dari MQTT. Setelah kode program selesai dibuat, kompres kode program tersebut menjadi zip dan upload kode program tersebut ke Amazon Web Service Lambda dengan cara buka alamat <https://console.aws.amazon.com> pilih Lambda, lalu pilih create new function. Maka akan ditampilkan halaman seperti Gambar 5.47. Isikan data seperti gambar tersebut.

Author from scratch [Info](#)

Name  
AlexaSensorFunction

Runtime  
Node.js 6.10

Role  
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more about Lambda execution roles.](#)  
Choose an existing role

Existing role  
You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.  
lambda\_basic\_execution

**Gambar 5. 36 Konfigurasi Amazon Web Service Lambda**

Selanjutnya klik create function pada pojok kanan bawah halaman website. Lalu halaman website akan diarahkan ke halaman selanjutnya. Pada halaman tersebut lakukan upload program dengan pilihan seperti Gambar 5.48.

Function code

Code entry type  
Upload a .ZIP file

Function package\*  
**Upload**  
For files larger than 10 MB, consider uploading via S3.

**Gambar 5. 37 Upload Kode Program**

## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini membahas mengenai pengujian dan analisis hasil dari implementasi sistem yang telah diterapkan melalui tahapan perancangan dan implementasi. Pengujian dilakukan untuk mengetahui apakah kebutuhan sistem telah terpenuhi. Pengujian dibagi menjadi beberapa tahap sesuai dengan tujuannya agar lebih mudah dalam proses analisa.

### 6.1 Pengujian Raspberry Pi sebagai modul Speech Recognition

#### 6.1.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui keberhasilan dari perangkat raspberry pi sebagai modul speech recognition. Tingkat dari keberhasilannya adalah ketika perangkat raspberry dapat merespon ketika user mengucapkan wake word **Alexa** dan *invocation name* dari skill yang telah di implementasikan pada Alexa Skills Kit, yaitu **Home Security**.

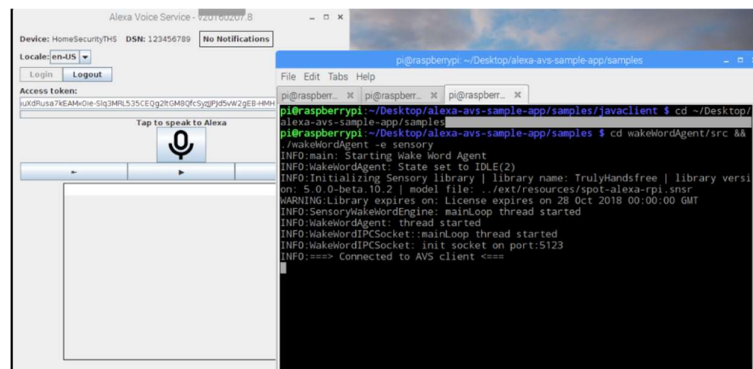
#### 6.1.2 Prosedur Pengujian

Prosedur yang akan dilakukan pada pengujian ini adalah sebagai berikut :

1. Perangkat Raspberry sudah terkoneksi dengan jaringan internet
2. Sudah melakukan tahap-tahap implementasi Sub Sistem *Microphone*, yaitu mengaktifkan *Alexa Web Service*, *Sample App*, dan *Word Engine*.
3. Setelah mengucapkan kata **Alexa**, tunggu sampai terdengar suara notifikasi yang menandakan sistem dalam kondisi *listening*.
4. Mengucapkan *Invocation Name*, yaitu **Home Security**

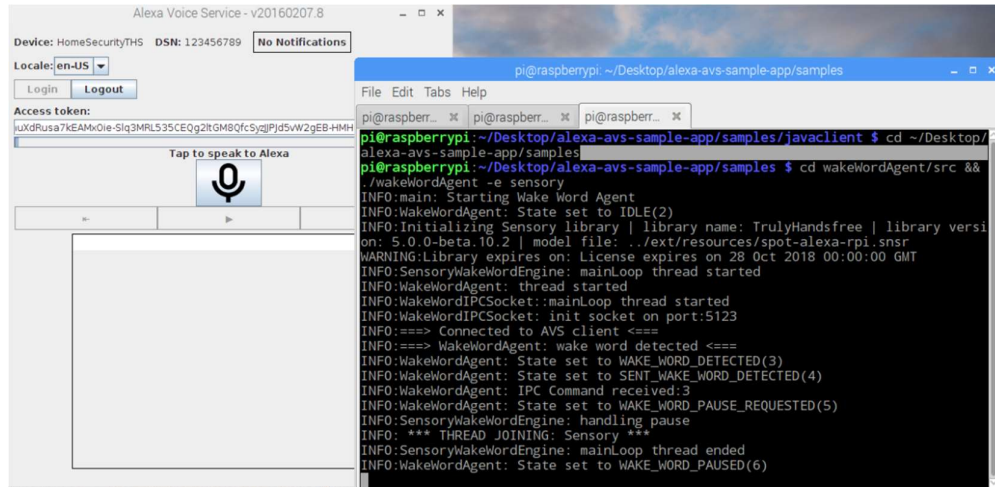
#### 6.1.3 Hasil Pengujian

Hasil dari pengujian ini, akan ditampilkan pada terminal raspberry pi yang sedang bertindak untuk menjalankan *Wake Word Engine* dan *Alexa Voice Service Sample App*. Gambar 6.1 menampilkan terminal dan sample app ketika pertama kali dijalankan.



Gambar 6. 1 Tampilan Terminal Raspberry dan Sample app

Selanjutnya adalah ketika wake word dan invocation name diucapkan, respon dari sistem dapat dilihat pada gambar 6.2 berikut. Pada sample app terlihat indicator diatas gambar speaker bergerak. Hal tersebut menandakan bahwa alexa sample app sedang listening.



**Gambar 6. 2 Wake word dan invocation name dikenali**

Pada tahap pengujian ini dilakukan pengucapan wake word dan invocation name sebanyak 10 kali. Hasilnya dapat dilihat pada table 6.1

**Tabel 6. 1 Hasil Percobaan Modul Speech Recognition**

Percobaan Ke-	Wake Word Terdeteksi	Invocation Name terdeteksi	Berhasil Terhubung ke Skill
1	ya	ya	ya
2	ya	ya	tidak
3	ya	ya	ya
4	ya	ya	ya
5	ya	ya	ya
6	ya	tidak	tidak
7	ya	ya	ya
8	ya	ya	ya
9	ya	ya	ya
10	ya	ya	ya

Penghitungan pengiriman data :  $(9:10) \times 100\% = 90\%$

#### 6.1.4 Analisa Pengujian

Berdasarkan data yang telah diperoleh dari hasil pengujian, data tersebut menunjukkan bahwa sub sistem mikrofone yang bertindak sebagai modul speech dapat berfungsi sebagai mana mestinya dengan tingkat keberhasilan 90 %, sehingga sub sistem berhasil tersambung ke alexa skills kit dan alexa voice service.

## 6.2 Pengujian Pengiriman data pada Sub Sistem sensor Ultrasonik

### 6.2.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui persentase keberhasilan sub sistem sensor Ultrasonik dalam menjalankan fungsinya. Tingkat keberhasilannya adalah ketika sub sistem sensor ultrasonik dapat mengirimkan data ke MQTT Broker pada jarak maksimal dan data yang dikirimkan berhasil diterima oleh MQTT Broker.

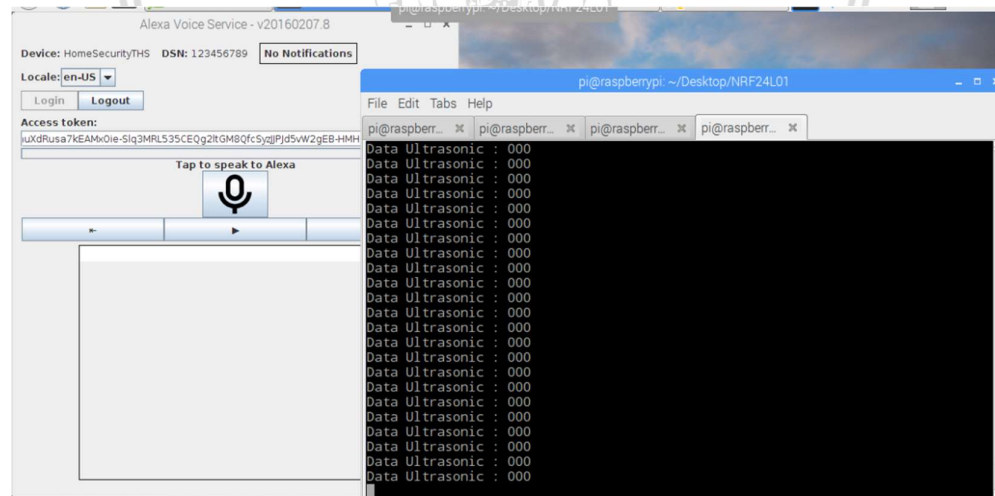
### 6.2.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini adalah sebagai berikut :

1. Perangkat Arduino telah terhubung dengan Modul nRF24L01 dan Sensor Ultrasonik.
2. Telah terhubung dengan sumber tegangan.
3. Perangkat Raspberry Pi telah terhubung dengan modul nRF24L01 dan sumber tegangan
4. Perangkat Raspberry telah terhubung dengan jaringan internet

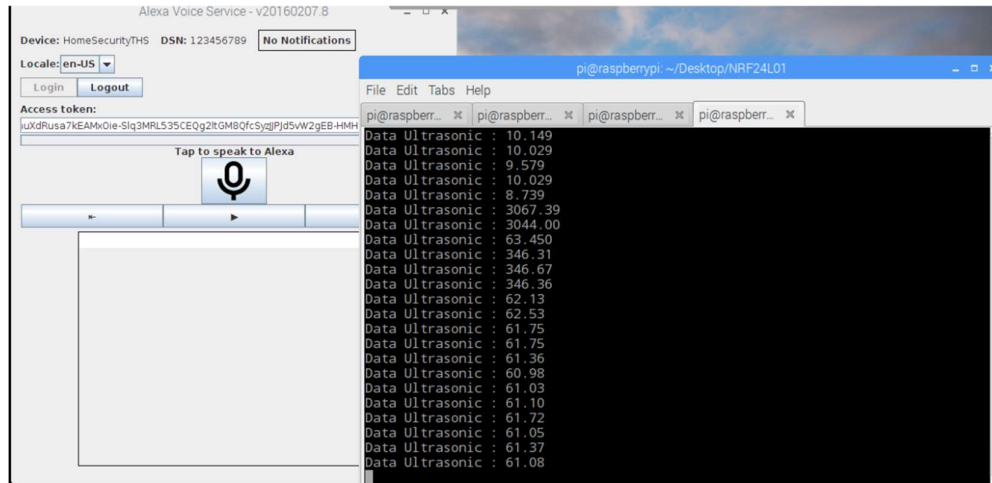
### 6.2.3 Hasil Pengujian

Hasil pengujian ini ditampilkan dalam terminal Raspberry Pi yang bertindak sebagai Receiver dan MQTT Broker. Gambar 6.3 dan Gambar 6.4 merupakan tampilan terminal raspberry pi ketika menerima paket data dari Sub Sistem Sensor Ultrasonik.

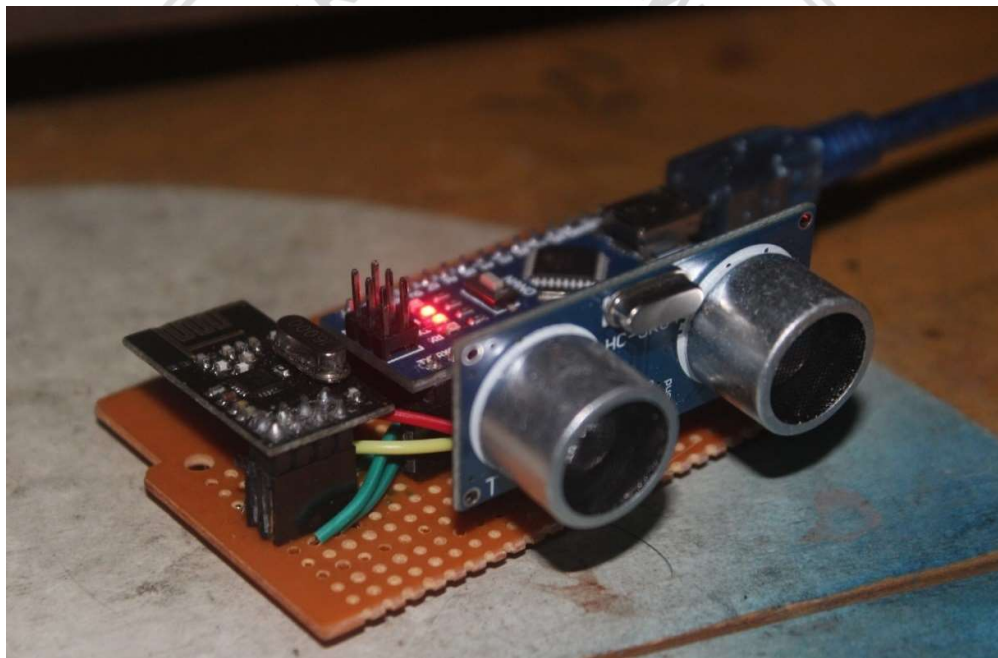


Gambar 6. 3 Hasil pengiriman data sensor menggunakan *obstacle*





**Gambar 6. 4 Hasil pengiriman data sensor tanpa *Obstacle***



**Gambar 6. 5 Ketika Sub Sistem Sensor Ultrasonik tidak dihadapkan obstacle**

Pada tahap pengujian ini, dilakukan percobaan untuk mengirimkan data dari sub sistem sensor ultrasonik sebanyak 18 kali untuk mengetahui jarak maksimal sub sistem ini dapat mengirimkan data ke MQTT Broker/Receiver.



**Tabel 6. 2 Hasil mengirimkan data pada jarak yang berbeda-beda**

percobaan ke -	Jarak (m)	pengiriman data	
		tanpa obstacle	menggunakan obstacle
1	1m	ya	ya
2	2m	ya	ya
3	3m	ya	ya
4	4m	ya	ya
5	5m	ya	ya
6	6m	ya	ya
7	7m	ya	ya
8	8m	ya	ya
9	9m	ya	ya
10	10m	ya	ya
11	11m	ya	ya
12	12m	ya	ya
13	13m	ya	ya
14	14m	ya	ya
15	15m	ya	ya
16	16m	ya	Ya

Penghitungan pengiriman data :  $(16:16) \times 100\% = 100\%^*$

### 6.2.4 Analisa Pengujian

Data dari hasil percobaan mengirimkan data dari sub sistem sensor ultrasonik ke MQTT Broker/Receiver memiliki tingkat keberhasilan 100 % didalam jarak 16 meter. Hal tersebut dapat dilihat pada Tabel 6.2 yaitu kesesuaian antara jarak dengan keberhasilan pengiriman data.

## 6.3 Pengujian Pengiriman data Sub Sistem Sensor *PIR*

### 6.3.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui presentase keberhasilan sub sistem sensor *PIR* dalam mengirimkan data dari Arduino sebagai Sub Sistem/Transceiver menuju ke Raspberry Pi sebagai MQTT Broker/Receiver dan keberhasilan mendeteksi pergerakan dalam jarak tertentu.

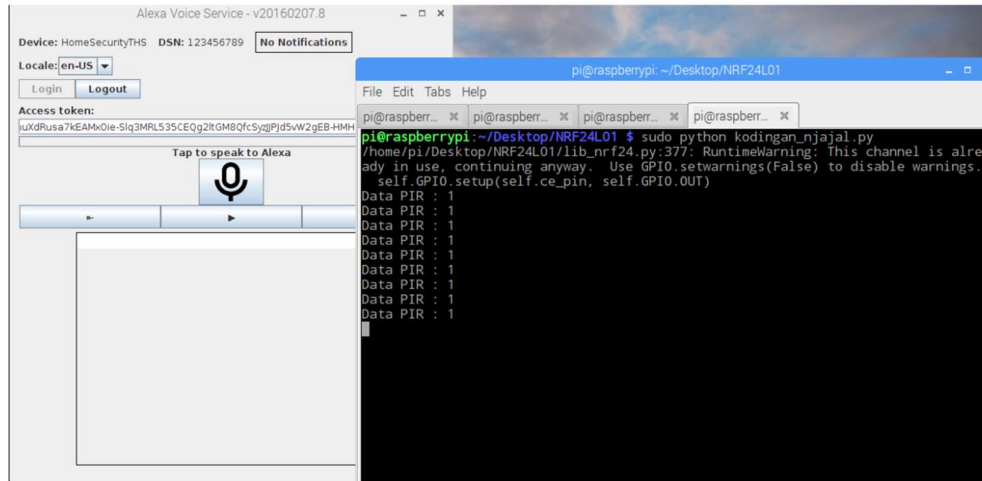
### 6.3.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini adalah sebagai berikut :

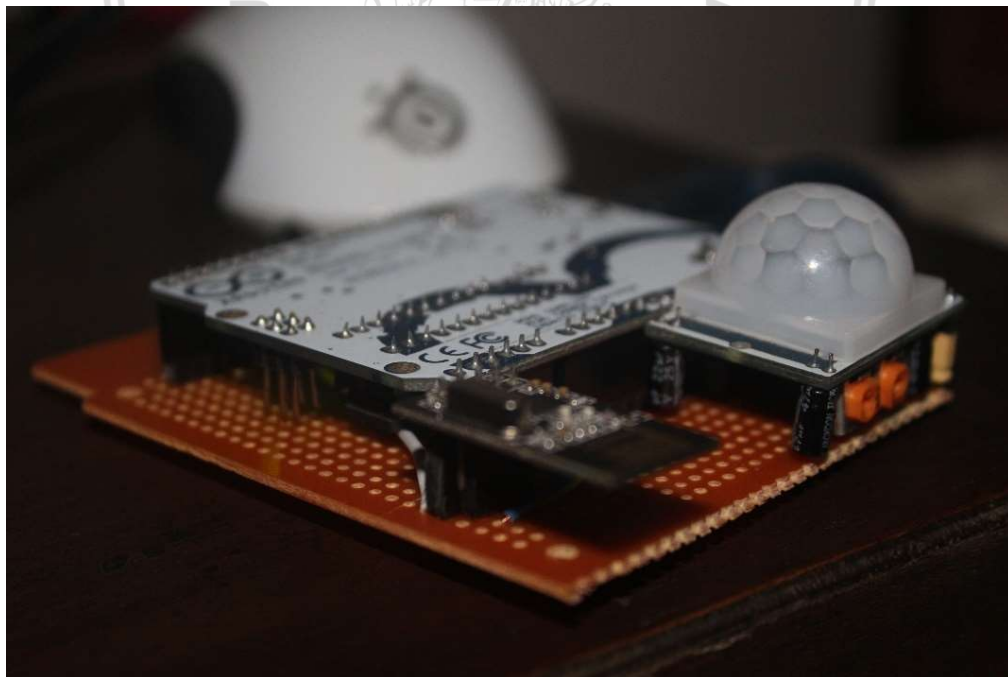
1. Perangkat Arduino telah terhubung dengan modul nRF24L01 dan sensor *PIR*
2. Perangkat Arduino telah terhubung dengan sumber tegangan.
3. Perangkat Raspberry Pi telah terhubung dengan modul nRF24L01 dan sumber tegangan
4. Perangkat Raspberry telah terhubung dengan jaringan internet

### 6.3.3 Hasil Pengujian

Hasil pengujian ini ditampilkan dalam terminal Raspberry Pi yang bertindak sebagai Receiver dan MQTT Broker. Gambar 6.6 merupakan tampilan terminal raspberry pi ketika menerima paket data dari Sub Sistem Sensor *PIR*.



Gambar 6. 6 Hasil Pengiriman data Sub Sistem Sensor *PIR*



Gambar 6. 7 Ketika Sub Sistem Sensor *PIR* mendeteksi pergerakan

Pada tahap pengujian ini, dilakukan percobaan untuk mengirimkan data dari sub sistem sensor ultrasonik sebanyak 31 kali untuk mengetahui jarak maksimal sub sistem ini dapat mengirimkan data ke MQTT Broker/Receiver.

**Tabel 6. 3 Hasil Pengiriman data pada Sub Sistem Sensor PIR**

Percobaan ke -	Jarak (m)	Berhasil mengirimkan data	Pergerakan terdeteksi
1	0,03m	ya	ya
2	0,05m	ya	ya
3	0,08m	ya	ya
4	0,09m	ya	ya
5	0,1m	ya	ya
6	0,15m	ya	ya
7	0,2m	ya	ya
8	0,25m	ya	ya
9	0,3m	ya	ya
10	0,34m	ya	ya
11	0,36m	ya	ya
12	0,4m	ya	ya
13	0,5m	ya	ya
14	1 m	ya	tidak
15	1,5 m	ya	tidak
16	2 m	ya	tidak
17	2,5 m	ya	tidak
18	3 m	ya	tidak
19	3,5 m	ya	tidak
20	4 m	ya	tidak
21	4,5 m	ya	tidak
22	4,8 m	ya	tidak
23	5 m	ya	tidak
24	6m	ya	tidak
25	7,5m	ya	tidak
26	9m	ya	tidak
27	12m	ya	tidak
28	13,5m	ya	tidak
29	15m	ya	tidak
30	15,5m	ya	tidak

Penghitungan jarak pengiriman data:  $(30:30) \times 100\% = 100\%*$

Penghitungan sensor mendeteksi gerakan :  $(12 : 30) \times 100\% = 40\%$

### 6.3.4 Analisa Pengujian

Data dari hasil percobaan mengirimkan data dari sub sistem sensor *PIR* ke MQTT Broker/Receiver memiliki tingkat keberhasilan 100%. Hal tersebut dapat dilihat pada Tabel 6.3 yaitu kesesuaian antara jarak dengan keberhasilan pengiriman data. Untuk mendeteksi gerak, sensor *PIR* memiliki tingkat keberhasilan 40%.

## 6.4 Pengujian pengiriman data pada Sub Sistem sensor *Microwave*

### 6.4.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui presentase keberhasilan sub sistem sensor *Microwave* dalam mengirimkan data dari Arduino sebagai Sub Sistem/Transceiver menuju ke Raspberry Pi sebagai MQTT Broker/Receiver dan keberhasilan mendeteksi pergerakan dalam jarak tertentu.

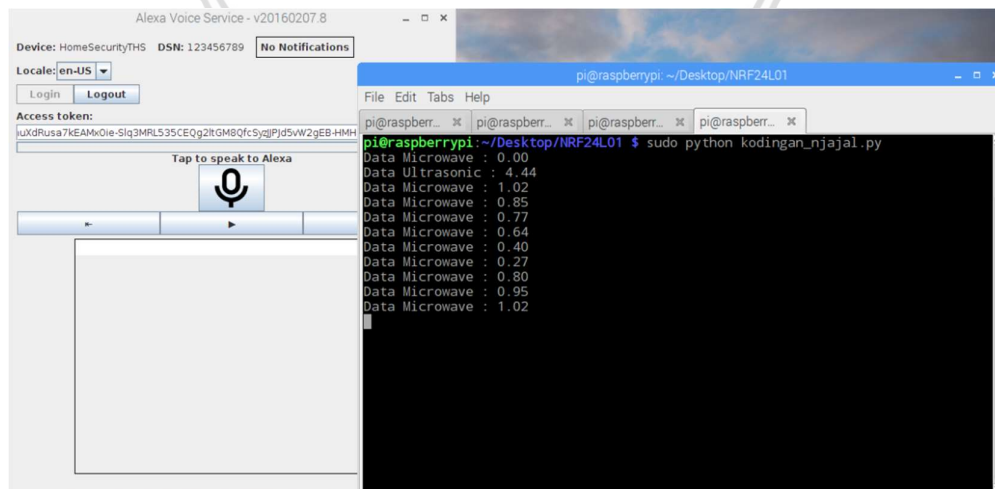
### 6.4.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini adalah sebagai berikut :

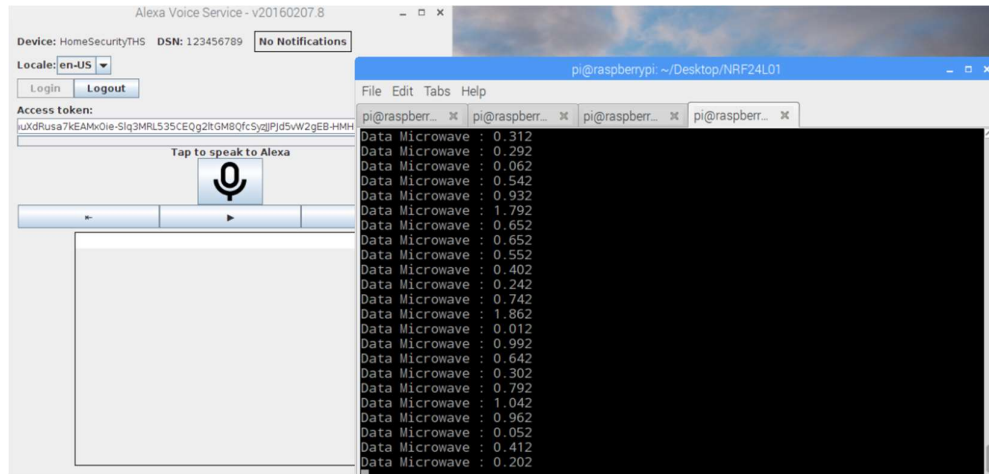
1. Sensor *Microwave* terhubung dengan rangkaian penguat sinyal LM324.
2. Perangkat Arduino telah terhubung dengan modul nRF24L01 dan Rangkaian sensor *Microwave*.
3. Perangkat Arduino telah terhubung dengan sumber tegangan.
4. Perangkat Raspberry Pi telah terhubung dengan modul nRF24L01 dan sumber tegangan.
5. Perangkat Raspberry telah terhubung dengan jaringan internet.

### 6.4.3 Hasil Pengujian

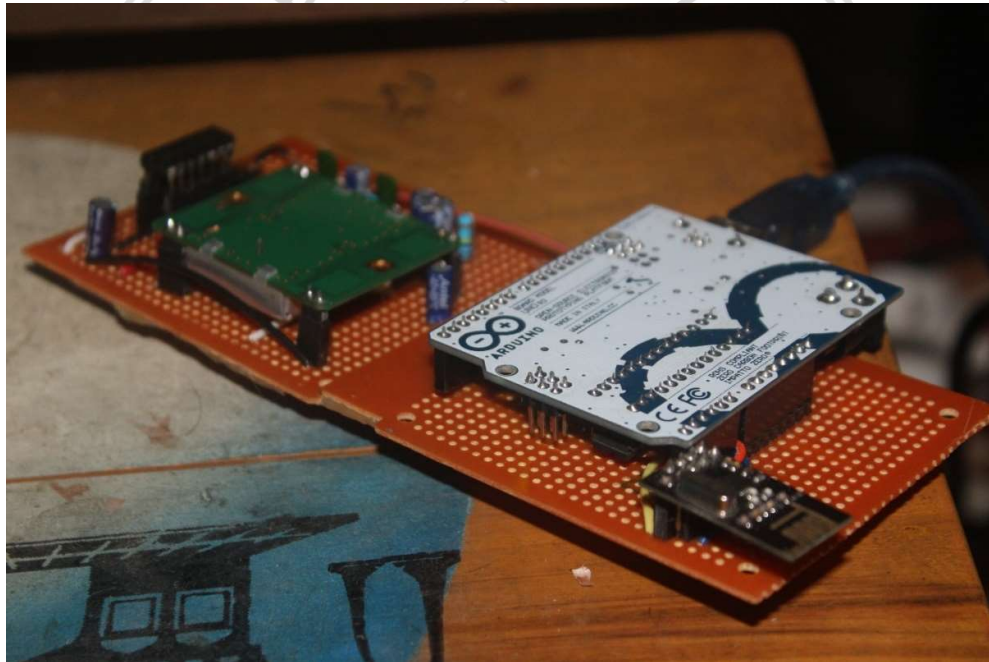
Hasil pengujian ini ditampilkan dalam terminal Raspberry Pi yang bertindak sebagai Receiver dan MQTT Broker. Gambar 6.8 dan Gambar 6.9 merupakan tampilan terminal raspberry pi ketika menerima paket data dari Sub Sistem Sensor *Microwave*.



Gambar 6. 8 Hasil pengiriman data sensor *Microwave* pada jarak 5m.



Gambar 6. 9 Hasil pengiriman data sensor *Microwave* pada jarak 16m.



Gambar 6. 10 Sub Sistem Sensor *Microwave*

Pada tahap pengujian ini, dilakukan percobaan untuk mengirimkan data dari sub sistem sensor *microwave* sebanyak 28 kali untuk mengetahui jarak maksimal sub sistem ini dapat mengirimkan data ke MQTT Broker/Receiver.

**Tabel 6. 4 Hasil Pengiriman data pada Sub Sistem Sensor *Microwave***

Percobaan ke -	Jarak (m)	Pengiriman data		Pergerakan terdeteksi
		Tanpa obstacle	Menggunakan obstacle	
1	0,03m	ya	ya	ya
2	0,05m	ya	ya	ya
3	0,08m	ya	ya	ya
4	0,09m	ya	ya	ya
5	0,1m	ya	ya	ya
6	0,15m	ya	ya	ya
7	0,2m	ya	ya	ya
8	0,4m	ya	ya	tidak
9	0,6m	ya	ya	tidak
10	0,8m	ya	ya	tidak
11	0,9m	ya	ya	tidak
12	1m	ya	ya	tidak
13	2m	ya	ya	tidak
14	3m	ya	ya	tidak
15	4m	ya	ya	tidak
16	5m	ya	ya	tidak
17	6m	ya	ya	tidak
18	7m	ya	ya	tidak
19	8m	ya	ya	tidak
20	9m	ya	ya	tidak
21	10m	ya	ya	tidak
22	11m	ya	ya	tidak
23	12m	ya	ya	tidak
24	13m	ya	ya	tidak
25	14m	ya	ya	tidak
26	15m	ya	ya	tidak
27	16m	ya	ya	tidak

Penghitungan jarak pengiriman data:  $(27:27) \times 100\% = 100\%*$

Penghitungan jarak pendeteksian gerak:  $(7:27) \times 100\% = 25,92\%*$

#### 6.4.4 Analisa Pengujian

Data dari hasil percobaan mengirimkan data dari sub sistem sensor *PIR* ke MQTT Broker/Receiver memiliki tingkat keberhasilan 100 % didalam jarak 16 meter. Hal tersebut dapat dilihat pada Tabel 6.4 yaitu kesesuaian antara jarak dengan keberhasilan pengiriman data. Untuk mendeteksi gerak, sensor *microwave* memiliki tingkat keberhasilan 25,92 %.



## 6.5 Pengujian pengiriman data pada Sub Sistem Sensor *Vibration*

### 6.5.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui presentase keberhasilan sub sistem sensor *Vibration* dalam mengirimkan data dari Arduino sebagai Sub Sistem/Transceiver menuju ke Raspberry Pi sebagai MQTT Broker/Receiver.

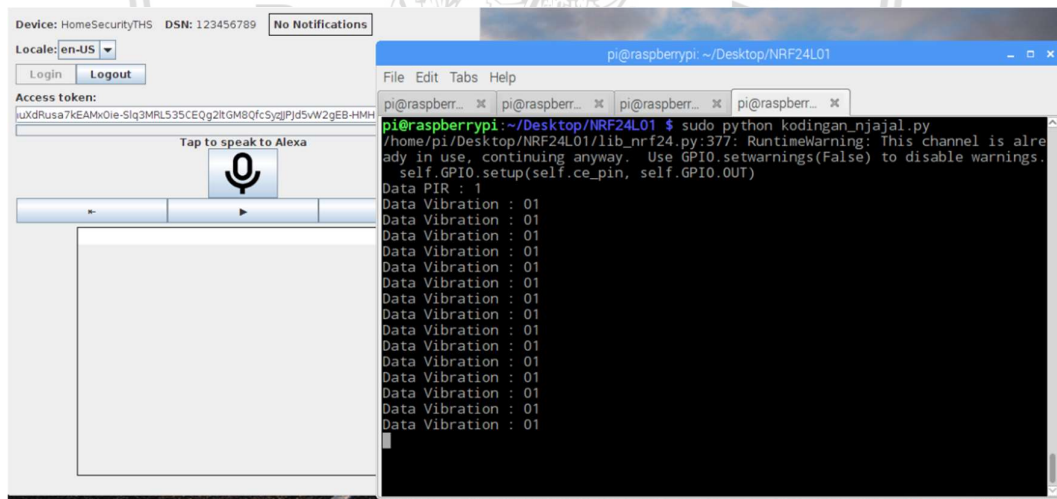
### 6.5.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini adalah sebagai berikut :

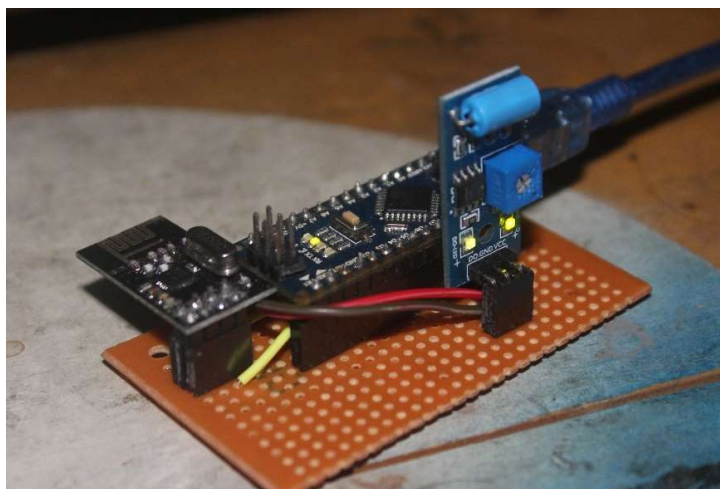
1. Perangkat Arduino telah terhubung dengan modul nRF24L01 dan sensor *vibration*.
2. Perangkat Arduino telah terhubung dengan sumber tegangan.
3. Perangkat Raspberry Pi telah terhubung dengan modul nRF24L01 dan sumber tegangan.
4. Perangkat Raspberry telah terhubung dengan jaringan internet.

### 6.5.3 Hasil Pengujian

Hasil pengujian ini ditampilkan dalam terminal Raspberry Pi yang bertindak sebagai Receiver dan MQTT Broker. Gambar 6.11 merupakan tampilan terminal raspberry pi ketika menerima paket data dari Sub Sistem Sensor *Vibration*.



Gambar 6. 11 Hasil pengiriman data Sub Sistem Sensor *Vibration*



**Gambar 6. 12 Sub Sistem Sensor *Vibration***

Pada tahap pengujian ini, dilakukan percobaan untuk mengirimkan data dari sub sistem sensor *vibration* sebanyak 13 kali untuk mengetahui jarak maksimal sub sistem ini dapat mengirimkan data ke MQTT Broker/Receiver.

**Tabel 6. 5 Hasil Pengiriman data pada Sub Sistem Sensor *Vibration***

Percobaan ke -	Jarak (m)	Berhasil mengirimkan data	Getaran terdeteksi
1	1 m	ya	ya
2	3m	ya	ya
3	6,5m	ya	ya
4	8,5m	ya	ya
5	9,5m	ya	ya
6	10m	ya	ya
7	11,5m	ya	ya
8	12m	ya	ya
9	13m	ya	ya
10	14,5m	ya	ya
11	15m	ya	ya
12	16m	ya	ya

Penghitungan jarak pengiriman data:  $(12:12) \times 100\% = 100\%*$

#### 6.5.4 Analisa Pengujian

Data dari hasil percobaan mengirimkan data dari sub sistem sensor *PIR* ke MQTT Broker/Receiver memiliki tingkat keberhasilan 100 % didalam jarak 16 meter. Hal tersebut dapat dilihat pada Tabel 6.5 yaitu kesesuaian antara jarak dengan keberhasilan pengiriman data.

## 6.6 Pengujian Membaca data Sensor pada Alexa

### 6.6.1 Tujuan Penelitian

Pengujian ini bertujuan untuk mengetahui presentase keberhasilan Alexa dalam membaca data sensor yang telah di subscribe dari MQTT dan untuk mengetahui presentase akurasi dari sistem dalam menjalankan fungsinya yang diucapkan oleh 5 orang yang berbeda. Pada tahap ini, semua data sensor telah terkumpul kemudian data sensor siap untuk di bacakan sebagai output suara.

### 6.6.2 Prosedur Pengujian

Prosedur yang dilakukan pada pengujian ini adalah sebagai berikut :

1. Perangkat raspberry telah terhubung ke internet.
2. Tahap-tahap pada implementasi Sub sistem mikrofon telah dilakukan yaitu mengaktifkan web service, sample app dan wake word engine.
3. Selanjutnya adalah ucapkan kata **alexa**, tunggu hingga notifikasi suara yang menandakan sistem sudah dalam kondisi listen.
4. Lalu ucapkan invocation name yaitu **home security**.
5. Selanjutnya ucapkan perintah untuk membaca sensor missal : “read *microwave* sensor”.

### 6.6.3 Hasil Pengujian

Hasil Pengujian ini berupa output suara. Tingkat keberhasilan hasil pengujian dapat dilihat pada tabel 6.6, sedangkan untuk tingkat akurasi saat sistem menjalankan fungsinya, dapat dilihat pada tabel 6.7.

**Tabel 6.6 Hasil Pengujian Alexa membaca data Sensor**

Percobaan Ke-	Wake Word Terdeteksi	Invocation Name terdeteksi	Berhasil Terhubung ke Skill	Alexa dapat membaca data sensor
1	ya	ya	ya	Ya
2	ya	ya	ya	Ya
3	ya	ya	ya	Ya
4	ya	ya	ya	Ya
5	ya	ya	ya	Ya
6	ya	ya	ya	Ya
7	ya	ya	ya	Ya
8	ya	ya	ya	Ya
9	ya	ya	ya	Ya
10	ya	ya	ya	Ya

**Tabel 6.7 Hasil Pengujian keseluruhan Sistem dalam menjalankan fungsinya**

Percobaan Ke-	Wake Word Terdeteksi	Invocation Name terdeteksi	Berhasil Terhubung ke Skill	Alexa dapat membaca data sensor	Speaker
1	Ya	Ya	ya	Ya	Speaker 1
2	Ya	Ya	Ya	Ya	Speaker 1
3	Ya	Ya	Ya	Ya	Speaker 2
4	Ya	Ya	Ya	Ya	Speaker 2
5	Ya	Ya	Ya	Ya	Speaker 3
6	Ya	Ya	Ya	Ya	Speaker 3
7	Ya	Ya	Ya	Tidak	Speaker 4
8	Ya	Ya	Ya	Ya	Speaker 4
9	Ya	Ya	Ya	Ya	Speaker 5
10	Ya	Ya	Ya	Tidak	Speaker 5

Penghitungan perintah suara ke Alexa:  $(8:10) \times 100\% = 80\%*$

#### 6.6.4 Analisa Pengujian

Data dari hasil percobaan uji tingkat akurasi menunjukan bahwa untuk perintah membaca data sensor dapat bekerja dengan tingkat keberhasilan 80 %. Hal tersebut dapat dilihat pada Tabel 6.6 dan Tabel 6.7.

## BAB 7 PENUTUP

Pada bab ini penulis memberikan kesimpulan berdasarkan perancangan, implementasi, pengujian, dan analisis terhadap sistem yang telah dilakukan. Selain itu penulis juga memberikan saran agar penelitian ini dapat dilanjutkan agar penelitian ini menjadi lebih baik dan berguna bagi pembaca. Berikut ini merupakan kesimpulan dan saran yang akan diberikan:

### 7.1 Kesimpulan

Berdasarkan perancangan, implementasi, pengujian, dan analisis yang telah dilakukan, maka penulis menyimpulkan:

1. Pengimplementasian Alexa Voice Service pada perangkat Raspberry Pi dilakukan dengan cara cloning file ke github alexa voice service, kemudian melakukan instalasi
2. Pada bagian Alexa Skill Kit dilakukan perancangan Intent Schema, Custom Slot, dan Sample Utterance supaya suara yang dikirimkan Raspberry Pi melalui Alexa Voice Service dapat diterjemahkan. Bahasa pemrograman yang digunakan untuk membuat Kode Program pada AWS Lambda adalah Node.js. Kode Program tersebut berisi handlers dari intent request yang dikirim oleh alexa skills kit.
3. Alexa dapat membaca data dari sensor *PIR*, sensor *microwave*, sensor *vibration*, dan sensor *ultrasonic* yang telah di *publish* ke MQTT dan kemudian di *subscribe* dari MQTT dengan cara mengubah tipe data yang didapatkan menjadi tipe data String. Ketika dilakukan Pengujian tingkat akurasi dengan 5 *speaker* berbeda, tingkat keberhasilan total dari perintah untuk membaca data sensor adalah 80%

### 7.2 Saran

Berikut beberapa saran yang penulis harapkan dalam pengembangan sistem pada penelitian ini:

1. Dapat menambahkan sistem database untuk menyimpan data-data sensor, sehingga alexa dapat mengakses data sensor beserta waktunya.
2. Membuat sistem dapat dipanggil kapan saja tanpa menggunakan wake word
3. Melakukan penelitian lebih lanjut tentang bagaimana respon sistem ketika terdapat banyak noise suara dilingkungan sekitar.
4. Dapat mengintegrasikan Alexa kedalam aplikasi Smartphone.

## DAFTAR PUSTAKA

- Ana, M., Ian. B., Alec, Z., Adrian, N., Reggie, C., 2015. *Home Automation Using Raspberry Pi through Siri Enabled Mobile Devices*. 8th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM).
- Anusuya, M.A., Katti, S.K., 2009. *Speech Recognition by Machine: A Review*. International Journal of Computer Science and Information Security, Maret.
- Arduino, 2017. *What is Arduino?*. [online]. Tersedia di: <<https://www.arduino.cc/en/Guide/Introduction/>> [Diakses 5 Maret 2018]
- Badan Pusat Statistik 2017. Statistik Kriminal 2017. [online]. Tersedia di: <<https://www.bps.go.id/publication/2017/12/22/197562b7ad0ced87c08fada5/statistik-kriminal-2017/>> [Diakses 9 April 2018]
- Grant, C., 2017. *What Is Alexa? What Is the Amazon Echo, and Should You Get One?*. [online]. Tersedia di: < <https://thewirecutter.com/reviews/what-is-alexa-what-is-the-amazon-echo-and-should-you-get-one/>> [Diakses 16 Juli 2018]
- Hidayat, S. dan Farid, S., 2015. *Scheduler and Voice Recognition on Home Automation Control System*. 3rd International Conference on Information and Communication Technology (ICoICT).
- Kjetil, H., 2017. *Corporate social responsibility*. [online]. Tersedia di: < <https://blog.nordicsemi.com/getconnected/voice-control-in-the-smarthome/>> [Diakses 7 September 2017]
- Nordic Semiconductor, 2017. nRF24L01. [online]. Tersedia di: < <https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01/>> [Diakses 15 Maret 2018]
- Panwar A., Singh A., Kumawat R., Jaidka S., Garg K., 2017. *Eyrie smart home automation using Internet of Things*, in Computing Conference, London, UK.
- Prapto, P., 2015. *Speech Recognition*. [online]. Tersedia di: < <https://praptoprasojo.wordpress.com/2015/11/13/speech-recognition/>> [Diakses 20 Februari 2018]
- Rajalakshmi A., dan Shahnaser H., 2017. *Internet of Things using Node-Red and Alexa*. 17th International Symposium on Communications and Information Technologies (ISCIT)
- Raspberry Pi Org, 2017. *GPIO Raspberry Pi Models A and B*. [online]. Tersedia di:



<<https://www.raspberrypi.org/documentation/usage/gpio/>> [Diakses 11 Juli 2018]

Santoso, H., 2015. Cara Kerja Sensor Ultrasonik, Rangkaian, & Aplikasinya. [online]. Tersedia di: <<https://www.elangsakti.com/2015/05/sensor-ultrasonik/>> [Diakses 8 Desember 2017]

Scheidtmann, Andre, 2017. *Is there a Catch All Intend?*. [online]. Tersedia di: <<https://github.com/alexa-js/alexa-app/issues/213>> [Diakses 11 Juli 2018]

Theory Circuit, 2017. SW-420 Vibration Sensor Arduino Interface. [online]. Tersedia di: <<http://www.theorycircuit.com/sw-420-vibration-sensor-arduino-interface/>> [Diakses 28 September 2017]

Vincent, R., David, M., David, D., Bruno, M., Laurent, D., Christopher, L., 2016. *The Smart Home Concept : our immediate future. IEEE E-Learning in Industrial Electronics.*

Wang, Y. dan Dong, P., 2016. *The design and implementation of the voice control system of smart home based on iOS. Proceedings of 2016 IEEE International Conference on Mechatronics and Automation.*

Zhen, C., dan Shum, P., 2017. *Voice activated smart home design and implementation. IEEE on 2nd International Conference on Frontiers of Sensor Technologies (ICFST)*



